Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR TP 96338

Multiblock grid generation

Part I: Elliptic grid generation methods for structured grids

S.P. Spekreijse, J.W. Boerstoel

DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF.		SECURITY CLASS.
	NLR TP 96338 U		Unclassified

ORIGINATOR
National Aerospace Laboratory NLR, Amsterdam, The Netherlands

TITLE
Multiblock grid generation
Part I: Elliptic grid generation methods for structured grids

PRESENTED AT
27th Computational Fluid Dynamics Course, at the Von Karman Institute (VKI) for fluid dynamics, Belgium, 25-29 March 1996.

AUTHORS	DATE	pp	ref
S.P. Spekreijse, J.W. Boerstoel	960523	42	22

DESCRIPTORS

Aerodynamic configurations	Minimal surface
Algortihms	Multiblock grids
Boundary conditions	Navier-Stokes equation
Computational fluid dynamics	Orthogonality
Elliptic differential equation	Poisson equation
Grid generation (mathematics)	Structured grids (mathematics)
Laplace-Beltrami equation	

ABSTRACT
The paper is organized as follows. In Section 2, the Laplace equations, the parameter space and the algebraic transformation are presented for domains in two dimensional physical space. The resulting Poisson equations are derived together with the appropriate expressions of the control functions. The relationship with other methods is explained. The discretization and solution of the nonlinear elliptic equations is discussed and also the orthogonalization of the grid at boundaries. Examples of grids in 2D domains are given. Surface grid generation on minimal surfaces is discussed in Section 3. It is shown that grid generation on minimal surface is in fact the same problem as grid generation in a domain in 2D physical space. llustrations of grids on minimal surfaces are given.
Surface grid generation on surfaces with a prescribed shape is treated in Section 4. It is assumed that such surfaces are parametrized and that the parametrization is a differentiable one-to-one mapping from a unit square onto the surface. The generated surface grids are independent of the parametrization. The solution method to generate the grids in the interior of parametrized surfaces is different from that used for minimal surfaces. It is much easier to solve the two linear elliptic partial differential equations defined by the Laplace-Beltrami equations directly, instead of interchanging the dependent and independent variables which leads to an nonlinear elliptic system of partial differential equations. An inversion problem must then be solved afterwards. Such a simple solution method is only possible for parametrized surfaces. This is due to the fact that an initial grid folding free surface grid on a parametrized surface can be easily generated because the given parametrization is one-to-one.

Contents

1 Introduction

A graphical interactive multi-block grid generator, called ENGRID, has been developed at NLR to construct multi-block structured grids for the computation of flows based on the Euler and Navier-Stokes equations. Flows can be computed about complete aircraft configurations including propulsion aircraft components [11, 13, 14, 15]. Advanced algebraic grid generation techniques are applied to construct the grids [13, 14]. Extensive use of ENGRID at NLR and Fokker has demonstrated that the applied techniques are fast and sufficiently robust to create grids for the simulation of flows based on the Euler equations. However, the applied techniques show too often grid folding when Navier-Stokes grids are generated in the interior of curved surfaces and blocks with complex shapes. Therefore more robust grid generation techniques with a minimum of grid tuning parameters were needed to construct such grids efficiently. For this purpose, new elliptic grid generation methods have been developed with a maximum of robustness and a minimum of grid tuning parameters. The methods have been incorporated into the ENGRID code and have been applied successfully to generate boundary conforming Navier-Stokes grids in blocks and block-faces with complex shapes. The new elliptic grid generation methods are the topic of this lecture.

Since the pioneering work of Thompson on elliptic grid generation it is known that systems of elliptic second-order partial differential equations produce the best possible grids in the sense of smoothness and grid point distribution. The systems of elliptic second-order partial differential equations are Poisson-type systems with control functions to be specified. The secret of each "good" elliptic grid is the method to compute the control functions [10].

Originally Thompson and Warsi introduced the Poisson systems by considering a curvilinear coordinate system which satisfies a system of Laplace equations and which is transformed to another coordinate system [1, 2]. Then this new coordinate system satisfies a system of Poisson equations with control functions completely specified by the transformation between the two coordinate systems. However Thompson did not use this approach for grid generation. Instead he proposed to use the Poisson system with control functions specified directly rather than through a transformation [1]. Since then the general approach is to compute the control functions at the boundary and to interpolate them from the boundaries into the field [1, 8, 9, 10].

The main disadvantage of such an approach is that it is then not possible to prove that the system of Poisson equations defines a one-to-one map so that the computed grids may contain grid folding.

In this paper we will show that also Thompson's and Warsi's original idea to define the control functions by a transformation can be used for grid generation. An important advantage of this approach is that the corresponding Poisson system defines a one-to-one map if the transformation is one-to-one. It will be shown that it is not difficult to construct appropriate one-to-one transformations. For this purpose, nonlinear transfinite algebraic transformations will be used.

We will apply this approach to generate boundary conforming grids in domains in 2D and 3D physical space and on minimal surfaces and parametrized surfaces in 3D physical space.

Thus the underlying concept of the proposed grid generation method is to use a composite mapping. The idea is to introduce a parameter (coordinate) system in the given domains and surfaces which only depends on their shape and not on the prescribed boundary grid point distribution. The parameters are defined as normalized arclength at the boundaries and each parameter obeys the Laplace equation in the interior of a domain or the Laplace-Beltrami

equation in the interior of a surface.

For 2D domains and surfaces, the parameter system can be interpreted as a differentiable one-to-one mapping from a unit square, called the parameter space, onto the 2D domain or surface. This mapping is called here the elliptic transformation. The parameter space is a unit cube for a domain in 3D physical space.

A nonlinear transfinite algebraic transformation is constructed to control grid point distributions. This transformation maps the computational space onto the parameter space. The computational space is also considered as a unit square for 2D domains and surfaces and as a unit cube for 3D domains. Grids in computational space are always uniform. The algebraic transformation depends on the prescribed boundary grid point distribution. The algebraic transformation is constructed in such a way that the mapping is also differentiable and one-to-one.

Thus the algebraic transformation maps the computational space onto the parameter space and an elliptic transformation maps the parameter space onto the domains or surfaces in physical space. The composition of these two mappings is a differentiable one-to-one mapping from computational space onto the domains or surfaces in physical space and has a nonvanishing Jacobian. The composite mapping defines the grid point distribution in the interior of the domains or surfaces.

Although the composite mapping is one-to-one, this does not imply that generated grids are always grid folding free, because the discrete equations may not share this robustness property [7]. But it is sure that grid folding will always disappear when the grid is refined. Furthermore, it is our experience that grids produced by the composite mapping are hardly ever folded even when Navier-Stokes type of grids are generated in domains or surfaces with complex shapes.

The elliptic transformation is independent of the prescribed boundary grid point distribution and may thus be considered as a property of the domain or surface itself. The algebraic transformation depends on the prescribed grid point distribution. As we will see, the interior grid point distribution in parameter space, generated by the algebraic transformation, is always a good reflection of the grid point distribution at the boundary of the parameter space. Therefore, the interior grid point distribution in the domains and surfaces is also a good reflection of the prescribed boundary grid point distribution. This is not the case for grids solely based on the system of Laplace equations. Then the inherent smoothness of the Laplace operator makes the grids evenly spaced in the interior (for example, a boundary layer will be blown up and completely disappear). Therefore, grid generators solely based on the system of Laplace equations are unusable in practice.

Thompson [1] and Warsi [2] have shown that the composite mapping obeys an elliptic Poisson system with control functions completely defined by the algebraic transformation. The number of control functions is 6 for 2D domains and surfaces, and 18 for 3D domains. In our case, the control functions are specified by the algebraic transformation only and it is therefore not needed to compute the control functions at the boundary and to interpolate them into the interior of the domains or surfaces, as is the case of all well known elliptic grid generation systems based on Poisson systems [1, 8, 9, 10].

Also new and more useful expressions for the control functions are derived in a short and elegant way which only depend on the algebraic transformation itself and not also on the inverse of this transformation (which occurs in the expressions used by Warsi and Thompson).

The computed grids are in general not orthogonal at the boundary. Sometimes, grid orthogonality is very much desired. It is shown that the algebraic transformation can be

redefined to obtain a grid which is orthogonal at the boundary.

The nonlinear elliptic Poisson equations are solved by Picard iteration. The linearized equations are solved by excellent black-box multigrid solvers for linear problems which are developed by P.M. de Zeeuw at C.W.I. [17, 18, 19].

The paper is organized as follows. In Section 2, the Laplace equations, the parameter space and the algebraic transformation are presented for domains in two dimensional physical space. The resulting Poisson equations are derived together with the appropriate expressions of the control functions.The relationship with other methods is explained. The discretization and solution of the nonlinear elliptic equations is discussed and also the orthogonalization of the grid at boundaries. Examples of grids in 2D domains are given.

Surface grid generation on minimal surfaces is discussed in Section 3. It is shown that grid generation on a minimal surface is in fact the same problem as grid generation in a domain in 2D physical space. Illustrations of grids on minimal surfaces are given.

Surface grid generation on surfaces with a prescribed shape is treated in Section 4. It is assumed that such surfaces are parametrized and that the parametrization is a differentiable one-to-one mapping from a unit square onto the surface. The generated surface grids are independent of the parametrization. The solution method to generate the grids in the interior of parametrized surfaces is different from that used for minimal surfaces. It is much easier to solve the two linear elliptic partial differential equations defined by the Laplace-Beltrami equations directly, instead of interchanging the dependent and independent variables which leads to a nonlinear elliptic system of partial differential equations. An inversion problem must then be solved afterwards. Such a simple solution method is only possible for parametrized surfaces. This is due to the fact that an initial grid folding free surface grid on a parametrized surface can be easily generated because the given parametrization is one-to-one. Illustrations of grids on parametrized surfaces are given.

Grid generation in 3D domains is treated in Section 5. The elliptic and algebraic transformation is defined. The resulting Poisson equations are derived together with the appropriate expressions of the control functions. The discretization and solution of the nonlinear elliptic equations is discussed and examples of grids in 3D domains are given.

Finally, concluding remarks are made in Section 6.

2 2D Grid Generation

2.1 Derivation of the 2D grid generation equations

Consider a simply connected bounded domain $\mathcal{D}$ in two dimensional space with Cartesian coordinates $\vec{x} = (x, y)^T$. Suppose that $\mathcal{D}$ is bounded by four edges E_1, E_2, E_3, E_4. Let (E_1, E_2) and (E_3, E_4) be the two pairs of opposite edges as shown in Fig.1.

Define the computational space $\mathcal{C}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Assume that a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed which maps the boundary of $\mathcal{C}$ one-to-one on the boundary of $\mathcal{D}$. This mapping defines the boundary grid point distribution. Assume that

- $\xi \equiv 0$ at edge E_1 and $\xi \equiv 1$ at edge E_2,
- $\eta \equiv 0$ at edge E_3 and $\eta \equiv 1$ at edge E_4.

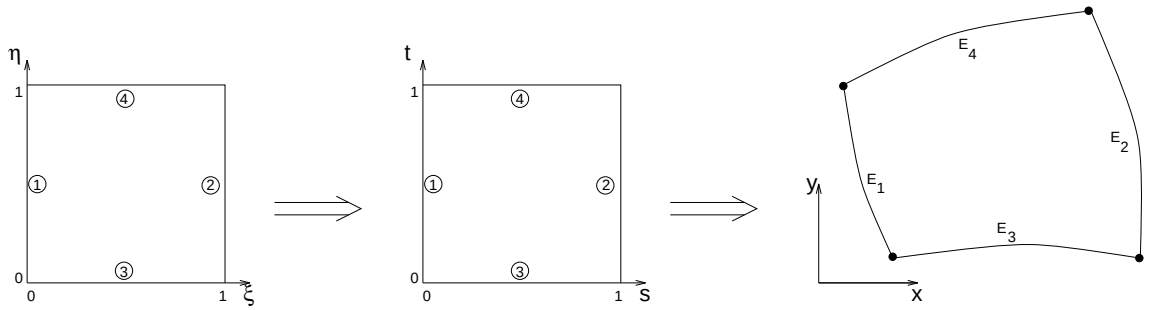

Figure 1: Transformation from computational (ξ, η) space to a domain $\mathcal{D}$ in Cartesian (x, y) space.

We wish to construct a mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ which obeys the boundary conditions and which is a differentiable one-to-one mapping. Furthermore, we require that the interior grid point distribution is a good reflection of the prescribed boundary grid point distribution.

A natural mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ exists which obeys these requirements. This mapping will be the composition of an algebraic transformation and an elliptic transformation based on the Laplace equations. The algebraic transformation is a differentiable one-to-one mapping from computational space $\mathcal{C}$ onto a parameter space $\mathcal{P}$. The parameter space is also a unit square. We will see below that the algebraic transformation will only depend on the prescribed boundary grid point distribution at the four edges of domain $\mathcal{D}$. The elliptic transformation is a differentiable one-to-one mapping from parameter space $\mathcal{P}$ onto domain $\mathcal{D}$. The elliptic transformation will only depend on the shape of domain $\mathcal{D}$ and is thus independent of the prescribed boundary grid point distribution. The elliptic transformation may thus be considered as a property of domain $\mathcal{D}$. The composition of these two mappings defines the interior grid point distribution and is a differentiable one-to-one mapping from computational domain $\mathcal{C}$ onto domain $\mathcal{D}$.

Introduce the parameter space $\mathcal{P}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s, t)^T$. Require that the parameters s and t obey:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2,
- s is the normalized arclength along edges E_3 and E_4.
- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4,
- t is the normalized arclength along edges E_1 and E_2.

Thus $\vec{s} : \partial\mathcal{D} \mapsto \partial\mathcal{P}$ is defined by these requirements. In the interior of $\mathcal{D}$ we require that s and t are harmonic functions of x and y, thus obey the Laplace equations:

$$\triangle s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} = s_{xx} + s_{yy} = 0, \tag{1}$$

$$\triangle t = \frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} = t_{xx} + t_{yy} = 0. \tag{2}$$

The two Laplace equations $\triangle s = 0$ and $\triangle t = 0$, together with the above specified boundary conditions, define the mapping $\vec{s} : \mathcal{D} \mapsto \mathcal{P}$. Note that this mapping only depends on the

shape of domain $\mathcal{D}$ and is independent of the prescribed boundary grid point distribution. By interchanging the dependent and independent variables, a non-linear elliptic partial differential equation can be derived for $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$. Thus we have to solve a non-linear elliptic boundary value problem in $\mathcal{P}$ in order to define this mapping. This mapping defines our elliptic transformation. It is well known that this mapping is differentiable and one-to-one [4].

The algebraic transformation must be a differentiable one-to-one mapping from computational space $\mathcal{C}$ onto the parameter space $\mathcal{P}$. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed and $\vec{x} : \partial\mathcal{P} \mapsto \partial\mathcal{D}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}$ is also defined.

From the preceding requirements it follows that

$$s(0,\eta) = 0\ ,\ s(1,\eta) = 1\ ,\ s(\xi,0) = s_{E_3}(\xi)\ ,\ s(\xi,1) = s_{E_4}(\xi), \tag{3}$$

where the functions s_{E_3}, s_{E_4} are monotonically increasing, and

$$t(\xi,0) = 0\ ,\ t(\xi,1) = 1\ ,\ t(0,\eta) = t_{E_1}(\eta)\ ,\ t(1,\eta) = t_{E_2}(\eta), \tag{4}$$

where the functions t_{E_1}, t_{E_2} are also monotonically increasing. Thus the four functions $t_{E_1}(\eta), t_{E_2}(\eta), s_{E_3}(\xi), s_{E_4}(\xi)$ are defined by the boundary grid point distribution.

The mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is now defined by the following two algebraic equations:

$$s = s_{E_3}(\xi)(1-t) + s_{E_4}(\xi)t, \tag{5}$$

$$t = t_{E_1}(\eta)(1-s) + t_{E_2}(\eta)s. \tag{6}$$

Eq.(5) implies that a coordinate line ξ = constant is mapped to the parameter space $\mathcal{P}$ as a straight line: s is a linear function of t, and Eq.(6) implies that a grid line η = constant is also mapped to $\mathcal{P}$ as a straight line: t is a linear function of s. For given values of ξ and η, the corresponding s and t values are found as the intersection point of the two straight lines. For this reason, the system defined by Eqs.(5),(6) is called the "algebraic straight line transformation" because of the use of straight lines in parameter space $\mathcal{P}$. It can be easily verified that this system defines a differentiable one-to-one mapping because of the positiveness of the Jacobian: $s_\xi t_\eta - s_\eta t_\xi > 0$.

The system defined by Eqs.(5),(6) can be interpreted as a transfinite interpolation with nonlinear blending functions and resembles the transfinite interpolation method of Soni [6].

The algebraic transformation $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ and the elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ are differentiable and one-to-one. Thus the composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ defined as $\vec{x}(\vec{\xi}) = \vec{x}(\vec{s}(\vec{\xi}))$ is also differentiable and one-to-one. Furthermore, due to the properties of the basic mappings, we may indeed expect that the interior grid point distribution will be a good reflection of the boundary grid point distribution.

In the remainder of this section, we will derive the set of non-linear elliptic partial differential equations which the composite mapping $\vec{x} = \vec{x}(\vec{s}(\vec{\xi}))$ has to fulfill.

It has already been noted by Warsi and Thompson that the composite mapping will obey an elliptic system of Poisson equations. However the system of Poisson equations as given in [1, 2] is not so useful because it contains control functions which depend also on the derivatives of the inverse mapping $\vec{\xi} : \mathcal{P} \mapsto \mathcal{C}$. It will be shown below that it is not difficult to obtain expressions for these control functions which only depend on the derivatives of the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ itself.

First introduce the two covariant base vectors

$$\vec{a}_1 = \frac{\partial \vec{x}}{\partial \xi} = \vec{x}_\xi \ , \ \vec{a}_2 = \frac{\partial \vec{x}}{\partial \eta} = \vec{x}_\eta, \tag{7}$$

and define the covariant metric tensor components as the inner product of the covariant base vectors

$$a_{ij} = (\vec{a}_i, \vec{a}_j) \ , \ i = \{1,2\}, j = \{1,2\}. \tag{8}$$

Then the contravariant base vectors $\vec{a}^1$ and $\vec{a}^2$ are defined according to the rules

$$(\vec{a}^i, \vec{a}_j) = \delta^i_j \ , \ i = \{1,2\}, j = \{1,2\}, \tag{9}$$

with δ^i_j the Kronecker symbol. Define the contravariant metric tensor components

$$a^{ij} = (\vec{a}^i, \vec{a}^j) \ , \ i = \{1,2\}, j = \{1,2\}, \tag{10}$$

so that

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} a^{11} & a^{12} \\ a^{12} & a^{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{11}$$

and

$$\vec{a}^1 = a^{11}\vec{a}_1 + a^{12}\vec{a}_2 \ , \ \vec{a}^2 = a^{12}\vec{a}_1 + a^{22}\vec{a}_2. \tag{12}$$

Introduce the determinant J^2 of the covariant metric tensor: $J^2 = a_{11}a_{22} - a_{12}^2$.

Now consider an arbitrary function $\phi = \phi(\xi, \eta)$. Then ϕ is also defined in domain $\mathcal{D}$ and the Laplacian of ϕ is expressed as

$$\triangle\phi = \phi_{xx} + \phi_{yy} = \frac{1}{J}\left\{\left(Ja^{11}\phi_\xi + Ja^{12}\phi_\eta\right)_\xi + \left(Ja^{12}\phi_\xi + Ja^{22}\phi_\eta\right)_\eta\right\}, \tag{13}$$

which may be found in every textbook on Tensor Analysis and Differential Geometry (for example see [21], page 227). Take as special cases respectively $\phi \equiv \xi$ and $\phi \equiv \eta$. Then Eq.(13) yields

$$\triangle\xi = \frac{1}{J}\left\{\left(Ja^{11}\right)_\xi + \left(Ja^{12}\right)_\eta\right\} \ , \ \triangle\eta = \frac{1}{J}\left\{\left(Ja^{12}\right)_\xi + \left(Ja^{22}\right)_\eta\right\}. \tag{14}$$

Thus the Laplacian of ϕ can also be expressed as

$$\triangle\phi = a^{11}\phi_{\xi\xi} + 2a^{12}\phi_{\xi\eta} + a^{22}\phi_{\eta\eta} + \triangle\xi\phi_\xi + \triangle\eta\phi_\eta. \tag{15}$$

Substitution of respectively $\phi \equiv s$ and $\phi \equiv t$ in this equation yields

$$\triangle s = a^{11}s_{\xi\xi} + 2a^{12}s_{\xi\eta} + a^{22}s_{\eta\eta} + \triangle\xi s_\xi + \triangle\eta s_\eta, \tag{16}$$

$$\triangle t = a^{11}t_{\xi\xi} + 2a^{12}t_{\xi\eta} + a^{22}t_{\eta\eta} + \triangle\xi t_\xi + \triangle\eta t_\eta. \tag{17}$$

Using these equations and the property that s and t are harmonic in domain $\mathcal{D}$, thus $\triangle s = 0$ and $\triangle t = 0$, we find the following expressions for the Laplacian of ξ and η

$$\begin{pmatrix} \triangle\xi \\ \triangle\eta \end{pmatrix} = a^{11}\vec{P}_{11} + 2a^{12}\vec{P}_{12} + a^{22}\vec{P}_{22}, \tag{18}$$

where

$$\vec{P}_{11} = -T^{-1} \begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix} \ , \ \vec{P}_{12} = -T^{-1} \begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix} \ , \ \vec{P}_{22} = -T^{-1} \begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \end{pmatrix}, \tag{19}$$

and the matrix T is defined as

$$T = \begin{pmatrix} s_\xi & s_\eta \\ t_\xi & t_\eta \end{pmatrix}. \tag{20}$$

The six coefficients of the vectors $\vec{P}_{11} = (P^1_{11}, P^2_{11})^T$, $\vec{P}_{12} = (P^1_{12}, P^2_{12})^T$ and $\vec{P}_{22} = (P^1_{22}, P^2_{22})^T$ are so called control functions. The six control functions are completely defined and easily computed for a given algebraic transformation $\vec{s} = \vec{s}(\vec{\xi})$. Different and less useful expressions of these control functions can also be found in [1, 2].

Finally, substitution of $\phi \equiv \vec{x}$ in Eq.(15) yields

$$\triangle\vec{x} = a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta} + \triangle\xi\vec{x}_\xi + \triangle\eta\vec{x}_\eta \ . \tag{21}$$

Substitution of Eq.(18) into this equation and using the fact that $\triangle\vec{x} \equiv 0$ we arrive at the following Poisson grid generation system

$$\begin{aligned} a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta} \quad &+ \quad \left(a^{11}P^1_{11} + 2a^{12}P^1_{12} + a^{22}P^1_{22}\right)\vec{x}_\xi \\ &+ \quad \left(a^{11}P^2_{11} + 2a^{12}P^2_{12} + a^{22}P^2_{22}\right)\vec{x}_\eta = 0. \end{aligned} \tag{22}$$

Using Eqs.(8),(11) we find the following well known expressions for the contravariant metric tensor components:

$$J^2a^{11} = a_{22} = (\vec{x}_\eta, \vec{x}_\eta) \ , \ J^2a^{12} = -a_{12} = -(\vec{x}_\xi, \vec{x}_\eta) \ , \ J^2a^{22} = a_{11} = (\vec{x}_\xi, \vec{x}_\xi). \tag{23}$$

Thus the Poisson grid generation system defined by Eq.(22) can be simplified by multiplication with J^2. Then we obtain:

$$\begin{aligned} a_{22}\vec{x}_{\xi\xi} - 2a_{12}\vec{x}_{\xi\eta} + a_{11}\vec{x}_{\eta\eta} \quad &+ \quad \left(a_{22}P^1_{11} - 2a_{12}P^1_{12} + a_{11}P^1_{22}\right)\vec{x}_\xi \\ &+ \quad \left(a_{22}P^2_{11} - 2a_{12}P^2_{12} + a_{11}P^2_{22}\right)\vec{x}_\eta = 0. \end{aligned} \tag{24}$$

This equation, together with the expressions for the control functions P^k_{ij} given by Eq.(19), forms our 2D grid generation system. Grids are computed by solving this quasi-linear system of elliptic partial differential equations with the prescribed boundary grid points as Dirichlet boundary conditions. The discretization of this Poisson system is described in Section 2.3.

2.2 Relationship with other methods

Suppose that the boundary grid point distribution is the same for both pairs of opposite edges of domain $\mathcal{D}$, thus $t_{E_1}(\eta) = t_{E_2}(\eta)$ and $s_{E_3}(\xi) = s_{E_4}(\xi)$. Then it follows from Eqs.(5),(6) that $s = s(\xi)$ and $t = t(\eta)$. The control functions become $P^1_{11} = P = -s_{\xi\xi}/s_\xi$, $P^2_{11} = 0$, $P^1_{12} = 0$, $P^2_{12} = 0$, $P^1_{22} = 0$, $P^2_{22} = Q = -t_{\eta\eta}/t_\eta$ and the Poisson grid generation system simplifies to

$$a_{22}(\vec{x}_{\xi\xi} + P\vec{x}_\xi) - 2a_{12}\vec{x}_{\xi\eta} + a_{11}(\vec{x}_{\eta\eta} + Q\vec{x}_\eta) = 0. \tag{25}$$

This is the common form of the Poisson system as used in the literature. The common approach is to compute the values of the two control functions P and Q from the boundary

grid point distribution and to interpolate these values in the interior.The values of P are computed at edges E_3 and E_4, the values of Q are computed at edges E_1 and E_2. For example consider edge E_1. Assume grid orthogonality, thus $(\vec{x}_\xi, \vec{x}_\eta) = 0$ i.e. $a_{12} = 0$. The Poisson system simplifies to

$$a_{22}(\vec{x}_{\xi\xi} + P\vec{x}_\xi) + a_{11}(\vec{x}_{\eta\eta} + Q\vec{x}_\eta) = 0. \tag{26}$$

Take the inner product of $\vec{x}_\eta$ with this equation, use $(\vec{x}_\xi, \vec{x}_\eta) = 0$ and ignore $(\vec{x}_{\xi\xi}, \vec{x}_\eta)$. Then the following expression is found for the boundary value of Q at edge E_1:

$$Q = -\frac{(\vec{x}_{\eta\eta}, \vec{x}_\eta)}{(\vec{x}_\eta, \vec{x}_\eta)}. \tag{27}$$

This is the computed boundary value of Q as used in the the method of Thomas and Middlecoff [8]. Along edge E_1 we have $\vec{x}_\eta = \vec{x}_t t_\eta$ and $\vec{x}_{\eta\eta} = \vec{x}_{tt} t_\eta^2 + \vec{x}_t t_{\eta\eta}$. Furthermore, t is defined as normalized arclength so that $(\vec{x}_t, \vec{x}_t) =$ constant and $(\vec{x}_t, \vec{x}_{tt}) = 0$. Thus $(\vec{x}_{\eta\eta}, \vec{x}_\eta) = (\vec{x}_t, \vec{x}_t) t_\eta t_{\eta\eta}$ so that Q is also equal to

$$Q = -\frac{t_{\eta\eta}}{t_\eta}. \tag{28}$$

Hence, for the special case that opposite boundary grid point distributions are the same, the method simplifies to the method of Thomas and Middlecoff.

2.3 Discretization and solution method

Consider a uniform rectangular grid of $(N+1) \times (M+1)$ points in computational space $\mathcal{C}$ defined as

$$\xi_{i,j} = \xi_i = i/N \ , \ \eta_{i,j} = \eta_j = j/M \ , \ i = 0 \ldots N, \ j = 0 \ldots M. \tag{29}$$

Assume that $\vec{x}_{i,j}$ is prescribed on the boundary of this grid and consider the computation of $\vec{x}_{i,j}$ in the interior of the computational grid based on the solution of the Poisson system defined by Eq.(24).

First we will compute the arclength normalized variables $s_{i,j}$ and $t_{i,j}$ based on the algebraic transformation defined by Eqs.(5),(6).

The arclength normalized variables $s_{i,j}$ and $t_{i,j}$ are computed at the boundary of the computational grid as follows. Compute the distance between succeeding points at the boundary:

$$\bar{d}_{0,j} = \| \vec{x}_{0,j} - \vec{x}_{0,j-1} \| \quad , \quad \bar{d}_{N,j} = \| \vec{x}_{N,j} - \vec{x}_{N,j-1} \| \ , \ j = 1 \ldots M, \tag{30}$$

$$\bar{d}_{i,0} = \| \vec{x}_{i,0} - \vec{x}_{i-1,0} \| \quad , \quad \bar{d}_{i,M} = \| \vec{x}_{i,M} - \vec{x}_{i-1,M} \| \ , \ i = 1 \ldots N. \tag{31}$$

Define the length of edges E_1, E_2, E_3, E_4 by

$$L_{E_1} = \sum_{j=1}^{M} \bar{d}_{0,j} \ , \ L_{E_2} = \sum_{j=1}^{M} \bar{d}_{N,j} \ , \ L_{E_3} = \sum_{i=1}^{N} \bar{d}_{i,0} \ , \ L_{E_4} = \sum_{i=1}^{N} \bar{d}_{i,M}, \tag{32}$$

and the normalized distances as

$$d_{0,j} = \bar{d}_{0,j}/L_{E_1} \quad , \quad d_{N,j} = \bar{d}_{N,j}/L_{E_2} \ , \ j = 1 \ldots M, \tag{33}$$

$$d_{i,0} = \bar{d}_{i,0}/L_{E_3} \quad , \quad d_{i,M} = \bar{d}_{i,M}/L_{E_4} \ , \ i = 1 \ldots N. \tag{34}$$

The arclength normalized variables $s_{i,j}$ and $t_{i,j}$ at the boundary are then defined by:

$$\begin{aligned} s_{0,j} &= 0\ ,\ s_{N,j} = 1\ ,\ j = 0 \ldots M, && (35)\\ t_{i,0} &= 0\ ,\ t_{i,M} = 1\ ,\ i = 0 \ldots N, && (36) \end{aligned}$$

and

$$\begin{aligned} s_{i,0} &= s_{i-1,0} + d_{i,0}\ ,\ s_{i,M} = s_{i-1,M} + d_{i,M}\ ,\ i = 1 \ldots N, && (37)\\ t_{0,j} &= t_{0,j-1} + d_{0,j}\ ,\ t_{N,j} = t_{N,j-1} + d_{N,j}\ ,\ j = 1 \ldots M. && (38) \end{aligned}$$

The arclength normalized variables $(s_{i,j}, t_{i,j})$ in the interior of the grid are now computed according to the algebraic straight line transformation defined by Eqs.(5),(6) and are thus found by solving simultaneously the two linear algebraic equations:

$$\begin{aligned} s_{i,j} &= s_{i,0}(1 - t_{i,j}) + s_{i,M} t_{i,j}, && (39)\\ t_{i,j} &= t_{0,j}(1 - s_{i,j}) + t_{N,j} s_{i,j}, && (40) \end{aligned}$$

for each pair $(i, j) \in (1 \ldots N-1, 1 \ldots M-1)$.

At each grid point (i, j), the six control functions P^1_{11},P^2_{11},P^1_{12}, P^2_{12},P^1_{22},P^2_{22} defined by Eq.(19), are now easily computed using central differences for the discretization of $s_{\xi\xi}$,$s_{\xi\eta}$,$s_{\eta\eta}$,s_ξ,s_η and $t_{\xi\xi}$,$t_{\xi\eta}$,$t_{\eta\eta}$,t_ξ,t_η.

Next, consider the iterative solution process of the nonlinear elliptic Poisson grid generation system defined by Eq.(24). Rewrite this system as

$$P\vec{x}_{\xi\xi} + 2Q\vec{x}_{\xi\eta} + R\vec{x}_{\eta\eta} + S\vec{x}_\xi + T\vec{x}_\eta = 0 \qquad (41)$$

with

$$\begin{aligned} P &= (\vec{x}_\eta, \vec{x}_\eta)\ ,\ Q = -(\vec{x}_\xi, \vec{x}_\eta)\ ,\ R = (\vec{x}_\xi, \vec{x}_\xi)\ ,\\ S &= PP^1_{11} + 2QP^1_{12} + RP^1_{22},\\ T &= PP^2_{11} + 2QP^2_{12} + RP^2_{22}. && (42) \end{aligned}$$

The solution of this system of nonlinear elliptic equations is obtained by Picard iteration:

$$P^{k-1}\vec{x}^k_{\xi\xi} + 2Q^{k-1}\vec{x}^k_{\xi\eta} + R^{k-1}\vec{x}^k_{\eta\eta} + S^{k-1}\vec{x}^k_\xi + T^{k-1}\vec{x}^k_\eta = 0 \qquad (43)$$

where k is the Picard index and

$$\begin{aligned} P^{k-1} &= (\vec{x}^{k-1}_\eta, \vec{x}^{k-1}_\eta)\ ,\ Q^{k-1} = -(\vec{x}^{k-1}_\xi, \vec{x}^{k-1}_\eta)\ ,\ R^{k-1} = (\vec{x}^{k-1}_\xi, \vec{x}^{k-1}_\xi)\ ,\\ S^{k-1} &= P^{k-1}P^1_{11} + 2Q^{k-1}P^1_{12} + R^{k-1}P^1_{22},\\ T^{k-1} &= P^{k-1}P^2_{11} + 2Q^{k-1}P^2_{12} + R^{k-1}P^2_{22}. && (44) \end{aligned}$$

Thus a current approximate solution

$$\vec{x}^{k-1} = \left\{ \vec{x}^{k-1}_{ij}, i = 0 \ldots N, j = 0 \ldots M \right\} \qquad (45)$$

is improved by the following steps:

- Compute the coefficients P^{k-1},Q^{k-1},R^{k-1},S^{k-1},T^{k-1} by applying central differences for the discretization of $\vec{x}_{\xi}^{k-1}$ and $\vec{x}_{\eta}^{k-1}$. Note that the six control functions remain unchanged during the iterative procedure.

- Discretize $\vec{x}_{\xi\xi}^{k}$, $\vec{x}_{\eta\eta}^{k}$, $\vec{x}_{\xi}^{k}$,$\vec{x}_{\eta}^{k}$ by using central differences. The discretization of the mixed derivative $\vec{x}_{\xi\eta}^{k}$ is done in a way as described in [20].

- After the discretization of $\vec{x}_{\xi\xi}^{k}$, $\vec{x}_{\xi\eta}^{k}$, $\vec{x}_{\eta\eta}^{k}$, $\vec{x}_{\xi}^{k}$, $\vec{x}_{\eta}^{k}$ we arrive at a linear system of equations for the unknowns $\vec{x}_{i,j}^{k}, i = 0 \ldots N, j = 0 \ldots M$ with Dirichlet boundary conditions. At each interior grid point (i, j) we have a nine-point stencil. This linear system is solved by the black-box multigrid solver MGD9V developed at C.W.I by P.M. de Zeeuw [17]. The multigrid solver MGD9V is called twice to compute the two components $x_{i,j}^{k}$ and $y_{i,j}^{k}$ of $\vec{x}_{i,j}^{k}$. The solution of the linear system provides a better approximate solution $\vec{x}^{k}$.

The complete process is repeated until a sufficiently accurate solution has been obtained. The initial start solution $\vec{x}^{0}$ is obtained by algebraic grid generation. The final grid is independent of the initial grid. Moreover, the quality of the initial grid is unimportant and severe grid folding of the initial grid is allowed. In general, about 10 Picard iterations are enough to obtain a sufficiently accurate solution of the nonlinear elliptic Poisson equations.

2.4 Orthogonality at boundaries

Grids obtained by the nonlinear elliptic Poisson grid generation system defined by Eq.(24) are grid folding free and have an excellent interior grid point spacing distribution. However, the computed grids are in general not orthogonal at the boundary and sometimes grids should be orthogonal at the boundary. Especially for Navier-Stokes computations, the orthogonality of the grid in a boundary layer is often desired.

Grid orthogonality at boundaries can be achieved as follows. Suppose that a grid has been computed based on the solution of the Poisson grid generation system with control functions specified by the algebraic straight-line transformation. Suppose that it is desired that the grid is orthogonal at all four edges of domain $\mathcal{D}$.

Redefine the elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ by imposing the following new set of boundary conditions for the harmonic functions s and t:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2,

- $\frac{\partial s}{\partial n} = 0$ along edges E_3 and E_4, where n is the outward normal direction,

- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4,

- $\frac{\partial t}{\partial n} = 0$ along edges E_1 and E_2, where n is the outward normal direction.

These new boundary conditions define a new mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$. Thus s is no longer the normalized arclength along edges E_3 and E_4, and t is no longer the normalized arclength along edges E_1 and E_2. It is not difficult to understand, by applying Gauss's integral formula for harmonic functions, that the Neumann boundary condition $\frac{\partial s}{\partial n} = 0$ implies that s cannot have a local extremum at edge E_3 and edge E_4 of domain $\mathcal{D}$. Similarly, t cannot have a local extremum at edge E_1 and edge E_2. Hence, s is still monotone along edges E_3 and E_4, and t is still monotone along edges E_1 and E_2.

The Neumann boundary conditions $\frac{\partial s}{\partial n} = 0$ along edges E_3 and E_4 also imply that a parameter line s = constant is a curve in domain $\mathcal{D}$ which is orthogonal at those edges. Similarly, a parameter line t = constant is a curve in $\mathcal{D}$ which is orthogonal at edge E_1 and edge E_2.

It is possible to compute the new harmonic functions s and t directly as functions of the computational coordinates (ξ, η) because of the existence of an initial mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$. We only have to solve two Laplace equations $\triangle s = 0$ and $\triangle t = 0$, together with the above specified combination of Dirichlet and Neumann boundary conditions, on an existing grid in domain $\mathcal{D}$. This is an elementary classical problem and the solution can be obtained rather easily. Some remarks about the discretization of the Laplace equation with Neumann boundary conditions are given at the end of this section.

Write the solution as $s = \tilde{s}(\xi, \eta)$ and $t = \tilde{t}(\xi, \eta)$. For our purposes, the only important information is the solution on the boundary. Redefine the edge functions by

$$s_{E_3}(\xi) = \tilde{s}(\xi, 0), s_{E_4}(\xi) = \tilde{s}(\xi, 1), t_{E_1}(\eta) = \tilde{t}(0, \eta), t_{E_2}(\eta) = \tilde{t}(1, \eta). \tag{46}$$

These new edge functions are still monotonically increasing.

The algebraic transformation $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is now redefined according to the following two algebraic equations:

$$s = s_{E_3}(\xi)H_0(t) + s_{E_4}(\xi)H_1(t), \tag{47}$$

$$t = t_{E_1}(\eta)H_0(s) + t_{E_2}(\eta)H_1(s). \tag{48}$$

where H_0 and H_1 are cubic Hermite interpolation functions defined as

$$H_0(s) = (1 + 2s)(1 - s)^2, H_1(s) = (3 - 2s)s^2, 0 \leq s \leq 1. \tag{49}$$

Note that $H_0(0) = 1$, $H_0'(0) = 0$, $H_0(1) = 0$, $H_0'(1) = 0$ and $H_1(0) = 0$, $H_1'(0) = 0$, $H_1(1) = 1$, $H_1'(1) = 0$. It follows from Eq.(47) that a coordinate line ξ = constant is mapped to parameter space $\mathcal{P}$ as a cubic curve which is orthogonal at both edge E_3 and edge E_4 in $\mathcal{P}$. Such a curve in parameter space $\mathcal{P}$ will thus be mapped by the new elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ as a curve which is orthogonal at both edge E_3 and edge E_4 in $\mathcal{D}$. Similar observations can be made for coordinate lines η = constant. Thus the grid will be orthogonal at all four edges in domain $\mathcal{D}$.

The composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ still obeys the Poisson grid generation system defined by Eq.(24). Thus the same system of elliptic equations can be solved to generate an orthogonal grid at the boundary. The only difference is that now $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is defined by Eqs.(47),(48) instead of Eqs.(5),(6).

Grid orthogonality at boundaries is obtained in three steps. First compute an initial grid based on the Poisson grid generation system with control functions specified according to the algebraic straight line transformation defined by Eqs.(5),(6). Next solve the two Laplace equations $\triangle s = 0$ and $\triangle t = 0$, together with the above specified combination of Dirichlet and Neumann boundary conditions, on this initial grid to obtain new edge functions $t_{E_1}(\eta)$, $t_{E_2}(\eta)$, $s_{E_3}(\xi)$, $s_{E_4}(\xi)$. Then recompute the grid based on the Poisson system but with control functions specified according to the algebraic transformation defined by Eqs.(47),(48).

Grid orthogonality at boundaries can introduce grid folding. Fortunately, grid folding will not easily arise. From Eq.(47) it follows that two different coordinate lines $\xi = \xi_1$,$\xi = \xi_2$, $\xi_1 \neq \xi_2$, are mapped to parameter space $\mathcal{P}$ as two disjunct cubic curves which are orthogonal

at both edge E_3 and edge E_4 in $\mathcal{P}$. This is due to the fact that $s_{E_3}(\xi)$ and $s_{E_4}(\xi)$ are monotonically increasing functions. The same holds for different coordinate lines $\eta = \eta_1$,$\eta = \eta_2$, $\eta_1 \neq \eta_2$. For given values of ξ and η, the corresponding s and t values are found as intersection point of two cubic curves. However, such two cubic curves can have more than one intersection point. In that case grid folding will occur. However, in practice we hardly ever encounter grid folding due to orthogonalization.

We have described a method to obtain an orthogonal grid at all four edges of domain $\mathcal{D}$. In practice, orthogonality of the grid is often only desired at one edge or two or three edges. Suppose for example that it is only desired to have an orthogonal grid at edge E_3. In that case, $t_{E_1}(\eta)$, $t_{E_2}(\eta)$ and $s_{E_4}(\xi)$ are defined as normalized arclength. Only $s_{E_3}(\xi)$ is computed by demanding that $\frac{\partial s}{\partial n} = 0$ along edge E_3 in $\mathcal{D}$. Thus only one Laplace equation $\triangle s = 0$ has to be solved to obtain $s_{E_3}(\xi)$ (with Dirichlet boundary conditions at edges E_1, E_2, E_4 and a Neumann boundary condition at edge E_3). Furthermore, it is sufficient that the algebraic transformation $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is such that a coordinate line $\eta =$ constant is mapped to $\mathcal{P}$ as a straight line and that a coordinate line $\xi =$ constant is mapped to $\mathcal{P}$ as a parabolic curve which is only orthogonal at edge E_3 in $\mathcal{P}$.

Finally we will show how the Laplace equation $\triangle s = 0$, together with the above specified boundary conditions, is discretized and solved on an existing grid in $\mathcal{D}$. The discretization and solution of $\triangle t = 0$ is obtained in the same way.

Consider a uniform rectangular grid of $(N+1) \times (M+1)$ points in computational space as defined by Eq.(29). Thus $\vec{x}_{i,j}$ is defined for all grid points (i, j). From Eq.(13) it follows that s obeys in computational space the linear second-order elliptic equation

$$\left(Ja^{11}s_\xi + Ja^{12}s_\eta\right)_\xi + \left(Ja^{12}s_\xi + Ja^{22}s_\eta\right)_\eta = 0 \tag{50}$$

which can be written in vector notation as

$$\text{div } (A \text{ grad } s) = 0 \tag{51}$$

where the matrix $A = A(\xi, \eta)$ is defined as

$$A = J \begin{pmatrix} a^{11} & a^{12} \\ a^{12} & a^{22} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{12} & a_{11} \end{pmatrix}. \tag{52}$$

At an interior grid point (i, j), the coefficients of matrix A can be directly computed by using central differences for $\vec{x}_\xi$ and $\vec{x}_\eta$. Thus Eq.(51) is a linear diffusion problem with given variable coefficients.

A finite-volume cell-centered approach is used to obtain the discretized equations. Integration of Eq.(51) on a control volume $\Omega \subset \mathcal{C}$ gives

$$\int_\Omega \text{div } (A \text{ grad } s) d\xi d\eta = \int_{\partial\Omega} (A \text{ grad } s, \vec{n}) d\sigma = \int_{\partial\Omega} (\text{ grad } s, A\vec{n}) d\sigma = 0 \tag{53}$$

where $\vec{n}$ is the outward unit normal vector and $d\sigma$ a line element. At an interior grid point (i, j) in $\mathcal{C}$, the discrete equation is derived in a straightforward way by applying Eq.(53) for a rectangular control volume $\Omega_{i,j}$ with sizes $1/N$ and $1/M$ around $(\xi_{i,j}, \eta_{i,j}) = (i/N, j/M)$. The result is a nine point stencil.

Half control volumes are used for boundary grid points. It is not difficult to show that the Neumann boundary condition $\frac{\partial s}{\partial n} = 0$ at a boundary in $\mathcal{D}$ transforms to (grad $s, A\vec{n}) = 0$

at the corresponding boundary in $\mathcal{C}$. Thus the flux is zero. This observation makes the discretization at boundary grid points also straightforward.

After discretization, we obtain at a linear system of equations for the unknowns $\{s_{i,j} \mid i = 0 \ldots N, j = 0 \ldots M\}$ with Dirichlet and Neumann boundary conditions. At each interior grid point (i, j) we have a nine-point stencil. This linear system is solved by the black-box multigrid solver MGD9V [17].

2.5 Illustrations

Examples of grids in 2D domains are shown in Figs.7,...,22. All grids are grid-folding free and the interior grid point distribution is a good reflection of the prescribed boundary grid point distribution. An initial grid (obtained with algebraic grid generation) is required as start solution for the nonlinear elliptic Poisson system. The final elliptic grid is independent of the initial grid. Moreover, the quality of the initial grid is unimportant and severe grid folding of the initial grid is allowed.

Fig.7 shows a region about a NACA0012 airfoil subdivided into four domains. The domains have common edges. The total number of edges is twelve. The boundary grid point distribution is prescribed at all twelve edges. Fig.8 shows a complete O-type Euler grid. Grid orthogonality is prescribed at the interior edges and at the boundary of the airfoil. A close-up near the airfoil of the domains and grid is shown in Fig.9 and Fig.10.

Fig.11 shows a region about a RAE2822 airfoil also subdivided into four domains. Again, the boundary grid point distribution is prescribed at all twelve edges and grid orthogonality is prescribed at the interior edges and at the boundary of the airfoil. Fig.12 shows a C-type Navier-Stokes grid. A close-up of the grid near the airfoil is shown in Fig.13.

A more complex example is a C-type Navier-Stokes grid around a wing with flap shown in Fig.14. Blow-up of the domain decomposition and multi-block grid are shown in Figs.15,...,18. Local grid refinement (see [16], Section 4.6) is applied near the wing and flap.

Fig.19 shows an initial grid around an complex artificial boundary with severe grid folding. This initial grid is obtained with an algebraic grid generation method. Fig.20 shows the Navier-Stokes grid around the complex artificial boundary obtained with the elliptic grid generation method. Grid orthogonality is prescribed. This grid illustrates the robustness of the elliptic grid generation method. Fig.21 and Fig.22 show details of the elliptic grid at respectively a convex and a concave part of the boundary. There is only some slight tendency that grid lines are more closely (widely) spaced near convex (concave) parts of the boundary.

3 Surface Grid Generation on Minimal Surfaces

Grid generation on a minimal surface is in fact a straightforward extension of grid generation in a domain in 2D physical space. Consider four connected curved edges situated in 3D physical space. A minimal surface is then defined as a surface bounded by these four edges and with zero mean curvature. Thus the shape of the minimal surface is a soap film bounded by the four curved edges. Again, a parameter system with two parameters is defined. The two parameters are normalized arclength at the four curved edges. Furthermore, it is required that both parameters obey the Laplace-Beltrami equation for surfaces. These two equations, together with the requirement that the mean curvature is identically zero, define a differentiable one-to-one mapping from parameter space (a unit square) onto the minimal surface. Thus this mapping is independent of the prescribed boundary grid point distribution at the four edges.

The same algebraic transformation as used for domains in 2D physical space is applied to map the computational space (a unit square) onto the parameter space.

We will now show that the set of non-linear elliptic partial differential equations which the composite mapping has to fulfill is the same Poisson system as defined by Eq.(24) but with $\vec{x} = (x, y, z)^T$ instead of $\vec{x} = (x, y)^T$. Thus grid generation on a minimal surface in 3D physical space is in fact equivalent to grid generation in a domain in 2D physical space. The result that a Poisson system of the form as defined by Eq.(24) can be used to compute a grid on a minimal surface can also be found as a special application of the formulas derived in [3].

As in the two dimensional case, consider again four curved edges E_1, E_2, E_3, E_4 but now situated in the three dimensional physical space with Cartesian coordinates $\vec{x} = (x, y, z)^T$. Let (E_1, E_2) and (E_3, E_4) be the two pairs of opposite edges as shown in Fig.2.

Introduce the parameter space $\mathcal{P}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s, t)^T$. Again require that the parameters s and t obey:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2,
- s is the normalized arclength along edges E_3 and E_4.
- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4,
- t is the normalized arclength along edges E_1 and E_2.

Furthermore, require that

$$\triangle s = 0, \tag{54}$$
$$\triangle t = 0, \tag{55}$$
$$H = 0, \tag{56}$$

where $\triangle$ is the Laplace-Beltrami operator for surfaces and H is the mean curvature.

These three requirements, together with the described boundary conditions define a unique mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{R}^3$. The shape of the surface defined by this mapping is a minimal surface because of the requirement that the mean curvature H is zero. The parametrization of the surface is defined by Eqs.(54),(55).

Define the minimal surface $\mathcal{S}$ as

$$\mathcal{S} = \{\vec{x}(s, t) \mid (s, t) \in \mathcal{P}\}. \tag{57}$$

Consider a prescribed boundary grid point distribution at the four edges E_1, E_2, E_3, E_4 of the minimal surface $\mathcal{S}$. Mathematically, the boundary grid point distribution can be defined as a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ where $\mathcal{C}$ is the computational space defined as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ is prescribed and $\vec{x} : \partial\mathcal{P} \mapsto \partial\mathcal{S}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}$ is also defined.

In exactly the same way as for the two dimensional case, the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is defined by the algebraic straight line transformation defined by Eqs.(5),(6). The mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{S}$ is defined by Eqs.(54),(55),(56). The composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{S}$ is defined as $\vec{x} = \vec{x}(\vec{s}(\vec{\xi}))$ and describes the interior grid point distribution on the minimal surface $\mathcal{S}$. Note that this composite mapping will be differentiable and one-to-one.

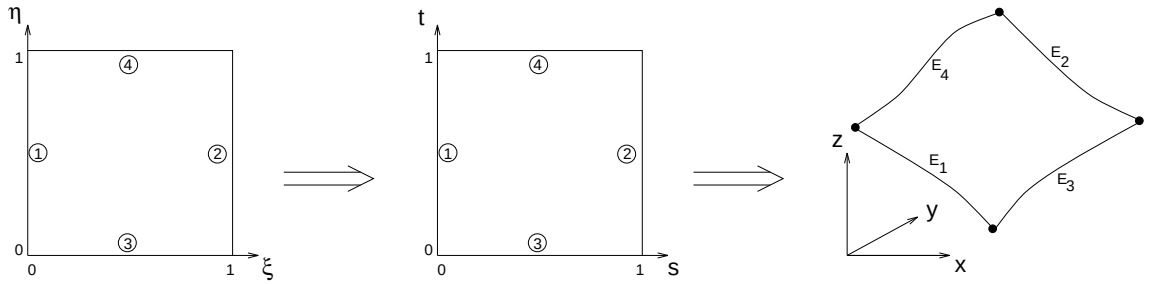

Figure 2: Transformation from computational (ξ, η) space to a minimal surface $\mathcal{S}$ in Cartesian (x, y, z) space.

What remains to be done is to derive the system of nonlinear elliptic partial differential equations which the composite mapping has to obey. Then the solution of this system defines the interior grid point distribution on the minimal surface $\mathcal{S}$.

For this purpose, introduce the two covariant base vectors

$$\vec{a}_1 = \vec{x}_\xi \ , \ \vec{a}_2 = \vec{x}_\eta. \tag{58}$$

The two covariant base vectors span the tangent plane of $\mathcal{S}$ at the corresponding point P. Define the unit surface normal as

$$\vec{n} = \frac{\vec{a}_1 \wedge \vec{a}_2}{\parallel \vec{a}_1 \wedge \vec{a}_2 \parallel}, \tag{59}$$

where $\wedge$ is the vector product operator. The contravariant base vectors $\vec{a}^1$ and $\vec{a}^2$ are defined according to the rules

$$(\vec{a}^i, \vec{a}_j) = \delta^i_j \ , \ i = \{1, 2\}, j = \{1, 2\}, \tag{60}$$

and

$$(\vec{a}^1, \vec{n}) = 0 \ , \ (\vec{a}^2, \vec{n}) = 0. \tag{61}$$

Thus the two contravariant base vectors are also lying in the tangent plane of $\mathcal{S}$ at the corresponding point P. Define the covariant metric tensor components by Eq.(8) and the contravariant metric tensor components by Eq.(10). Then Eqs.(11),(12) are still valid. Again introduce the determinant J^2 of the covariant metric tensor: $J^2 = a_{11}a_{22} - a_{12}^2$.

Now consider an arbitrary function $\phi = \phi(\xi, \eta)$. Then ϕ is also defined on surface $\mathcal{S}$ and the Laplace-Beltrami operator of ϕ is expressed as

$$\triangle\phi = \frac{1}{J}\left\{\left(Ja^{11}\phi_\xi + Ja^{12}\phi_\eta\right)_\xi + \left(Ja^{12}\phi_\xi + Ja^{22}\phi_\eta\right)_\eta\right\} \tag{62}$$

(see [21], page 227). As in the two-dimensional case, substitution of $\phi \equiv \xi$ and $\phi \equiv \eta$ into this equation yields Eq.(14). Thus the Laplace-Beltrami operator of ϕ can also be expressed as defined by Eq.(15). Substitution of respectively $\phi \equiv s$ and $\phi \equiv t$ in Eq.(15) and using the requirements expressed by Eqs.(54),(55) yields exactly the same expressions for $\triangle\xi$ and $\triangle\eta$ given by Eqs.(18),(19). Finally, substitution of $\phi \equiv \vec{x}$ in Eq.(15) yields Eq.(21).

The Laplace-Beltrami operator applied on $\vec{x}$ obeys a famous relation expressed by

$$\triangle\vec{x} = 2H\vec{n}, \tag{63}$$

where the mean curvature H is defined as

$$H = \frac{1}{2}\left(a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + a^{22}\vec{x}_{\eta\eta}, \vec{n}\right). \tag{64}$$

(for example see [22], Theorem 1, page 71). Using the requirement $H = 0$ yields

$$\triangle\vec{x} = 0. \tag{65}$$

Thus Eq.(18) and Eq.(21) with $\triangle\vec{x} = 0$ are also valid for minimal surfaces. Following the same derivation as given at the end of Section 2.1, we arrive at exactly the same nonlinear system of elliptic partial differential equations as expressed by Eq.(24). Thus an interior grid point distribution on a minimal surface is found by solving Eq.(24) with the prescribed boundary grid points as Dirichlet boundary conditions. The only difference compared to the two dimensional case is that now $\vec{x} = (x, y, z)^T$ instead of $\vec{x} = (x, y)^T$.

The same discretization and solution method as described in Section 2.3 can be used to solve the Poisson grid generation system in order to generate grids on minimal surfaces. The only difference compared to the two dimensional case is that three (instead of two) linear systems must be solved during one Picard iteration.

Grid orthogonality at boundaries can be obtained in the same way as described in Section 2.4.

One may ask whether it is useful to implement a method to compute grids on minimal surfaces in a 3D multi-block grid generator code. The answer is yes. Minimal surfaces may be used to define the geometry and grid for a block-face of which only the four face-edges are given. It is also possible to apply minimal surface grid generation when a grid must be generated in a block-face with four face-edges lying in a plane. Then the minimal surface is a plane surface bounded by the four edges. The grids in the 2D domains depicted in Figs.7,...,22 were generated in this way and are in fact grids on minimal surfaces.

An example of a grid on a characteristic minimal surface is shown in Fig.24. This is a so-called square Sherck surface [22]. The initial algebraic grid is shown in Fig.23. Fig.25 and Fig.26 illustrate what happens when the prescribed boundary grid point distribution is changed. These figures clearly show that the shape of the minimal surface is independent of the prescribed boundary grid point distribution.

4 Surface Grid Generation on Parametrized Surfaces

4.1 Derivation of the grid generation equations

In this section we develop a method to generate a grid on a parametrized surface which is independent of the parametrization. A generated grid only depends on the shape of the surface and the prescribed boundary grid point distribution at the four edges of the surface.

Consider a bounded surface $\mathcal{S}$ with a prescribed geometrical shape in three dimensional physical space with Cartesian coordinates $\vec{x} = (x, y, z)^T$. Assume that $\mathcal{S}$ is parametrized by a differentiable one-to-one mapping

$$\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}, \tag{66}$$

where $\mathcal{P}_{uv}$ is the unit square in two dimensional space with Cartesian coordinates $\vec{u} = (u, v)^T$. Define the four edges E_1, E_2, E_3, E_4 of surface $\mathcal{S}$ by

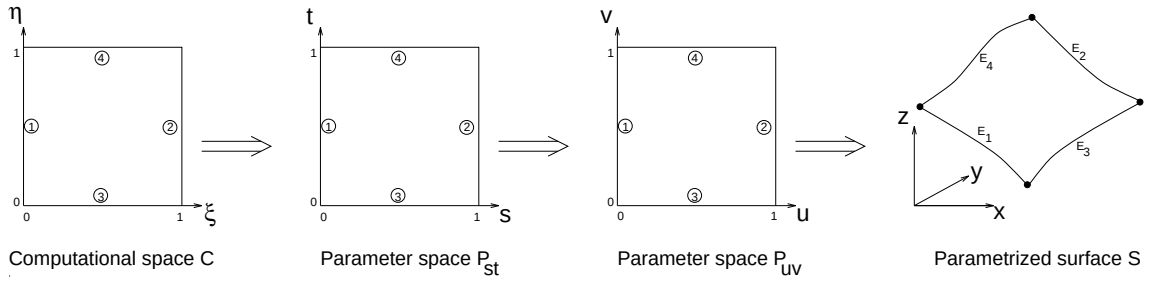

Figure 3: Transformation from computational (ξ, η) space to a parametrized surface $\mathcal{S}$ in Cartesian (x, y, z) space.

- $u \equiv 0$ at edge E_1 and $u \equiv 1$ at edge E_2,
- $v \equiv 0$ at edge E_3 and $v \equiv 1$ at edge E_4.

Thus (E_1, E_2) and (E_3, E_4) are the two pairs of opposite edges of surface $\mathcal{S}$ as shown in Fig.3. Introduce the parameter space $\mathcal{P}_{st}$ as the unit square in a two dimensional space with Cartesian coordinates $\vec{s} = (s, t)^T$. Again require that the parameters s and t obey:

- $s \equiv 0$ at edge E_1 and $s \equiv 1$ at edge E_2,
- s is the normalized arclength along edges E_3 and E_4,
- $t \equiv 0$ at edge E_3 and $t \equiv 1$ at edge E_4,
- t is the normalized arclength along edges E_1 and E_2.

Furthermore, require that $\triangle s = 0$ and $\triangle t = 0$ where $\triangle$ is the Laplace-Beltrami operator for surfaces. Hence the parameters s and t obey

$$\left(Ja^{11}s_u + Ja^{12}s_v\right)_u + \left(Ja^{12}s_u + Ja^{22}s_v\right)_v = 0, \tag{67}$$

$$\left(Ja^{11}t_u + Ja^{12}t_v\right)_u + \left(Ja^{12}t_u + Ja^{22}t_v\right)_v = 0, \tag{68}$$

where a^{ij} are the contravariant tensor components and J^2 is defined as the determinant of the covariant metric tensor. The contravariant tensor components a^{ij} are related to the covariant tensor components a_{ij} according to Eq.(11). The covariant metric tensor components are defined by Eq.(8), where the two covariant base vectors are now given by

$$\vec{a}_1 = \vec{x}_u \ , \ \vec{a}_2 = \vec{x}_v \ . \tag{69}$$

Thus the coefficients Ja^{11}, Ja^{12} and Ja^{22} in Eqs.(67),(68) are functions of u and v and Eqs.(67),(68) are therefore two uncoupled second-order linear partial differential equations for the functions $s = s(u, v)$ and $t = t(u, v)$.

Each boundary point of surface $\mathcal{S}$ has a unique (s, t) parameter value at $\partial\mathcal{P}_{st}$ and a unique (u, v) parameter value at $\partial\mathcal{P}_{uv}$. Thus each (u, v) parameter value at $\partial\mathcal{P}_{uv}$ has also a unique (s, t) parameter value at $\partial\mathcal{P}_{st}$. Thus the functions s and t are prescribed at the boundary of $\mathcal{P}_{uv}$. Hence, Eq.(67) together with the Dirichlet boundary conditions for s can be used to compute $s = s(u, v)$ and Eq.(68) together with the Dirichlet boundary conditions for t can be

used to compute $t = t(u, v)$. Only two linear partial differential equations have to be solved to define these mappings. These two mappings are compactly written as $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$. Note that $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$ is a differentiable one-to-one mapping so that the inverse mapping $\vec{u} : \mathcal{P}_{st} \mapsto \mathcal{P}_{uv}$ also exists.

Thus the composite mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$, defined as $\vec{x} = \vec{x}(\vec{u}(\vec{s}))$ also exists and is differentiable and one-to-one. Note that this mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ only depends on the shape of surface $\mathcal{S}$ and is independent of the original parametrization $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$. The mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ may thus be considered as a property of surface $\mathcal{S}$ and defines a new unique parametrization of $\mathcal{S}$.

Consider a prescribed boundary grid point distribution at the four edges E_1, E_2, E_3, E_4. Mathematically, the boundary grid point distribution can be defined as a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ where $\mathcal{C}$ is the computational space defined as the unit square in a two dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta)^T$. Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{S}$ is prescribed and $\vec{x} : \partial\mathcal{P}_{st} \mapsto \partial\mathcal{S}$ is defined as described above, it follows that $\vec{s} : \partial\mathcal{C} \mapsto \partial\mathcal{P}_{st}$ is also defined.

In exactly the same way as for the two dimensional case, the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}_{st}$ is now defined by the algebraic straight line transformation defined by Eqs.(5),(6). The composition of the mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}_{st}$ and the mapping $\vec{x} : \mathcal{P}_{st} \mapsto \mathcal{S}$ defines $\vec{x} : \mathcal{C} \mapsto \mathcal{S}$ and describes the interior grid point distribution on surface $\mathcal{S}$. Note that this composite mapping will also be differentiable and one-to-one.

Although it is possible to derive the system of nonlinear elliptic partial differential equations which the composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{S}$ has to obey, we prefer not to do so because it is much easier to solve the linear partial differential equations defined by Eqs.(67),(68) to define the mapping $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$ instead of interchanging the dependent and independent variables to obtain the nonlinear partial differential equations for the inverse mapping $\vec{u} : \mathcal{P}_{st} \mapsto \mathcal{P}_{uv}$. Thus the mapping $\vec{s} : \mathcal{P}_{uv} \mapsto \mathcal{P}_{st}$ is computed by solving Eqs.(67),(68) and an inversion problem is solved afterwards to compute the inverse mapping $\vec{u} : \mathcal{P}_{st} \mapsto \mathcal{P}_{uv}$.

This is possible due to the fact that the parametrization $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$ is one-to-one so that an initial grid folding free grid in surface $\mathcal{S}$ can be easily generated. Such an initial grid is obtained by applying the algebraic straight line algorithm in parameter space $\mathcal{P}_{uv}$. This is a different situation compared to grid generation in 2D domains or minimal surfaces where it is not possible to generate easily an initial grid folding free grid. Details of the solution method are described in the next section.

4.2 Discretization and solution method

Consider surface $\mathcal{S}$ with a prescribed boundary grid point distribution. Assume that there are $M + 1$ prescribed boundary grid points on edges E_1 and E_2, and $N + 1$ prescribed boundary grid points on edges E_3 and E_4. A boundary conforming grid in the interior of surface $\mathcal{S}$ is now obtained by the following algorithm.

step 1 Compute the corresponding boundary grid points in parameter space $\mathcal{P}_{uv}$. A boundary grid point $\vec{x}_B$ of surface $\mathcal{S}$ is related to a unique boundary grid point $\vec{u}_B$ of parameter space $\mathcal{P}_{uv}$ by the equation $\vec{x}(\vec{u}_B) = \vec{x}_B$ where $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$ is the given parametrization of surface $\mathcal{S}$. In practice, the corresponding parameter values $\vec{u}_B$ of a boundary grid point $\vec{x}_B$ are often already known.

step 2 Compute an initial grid $\vec{u}^I_{ij}$ by applying the algebraic straight line algorithm in parameter space $\mathcal{P}_{uv}$. Thus $\vec{u}^I_{ij}$ is computed according to Eqs.(39),(40) with s_{ij} and t_{ij}

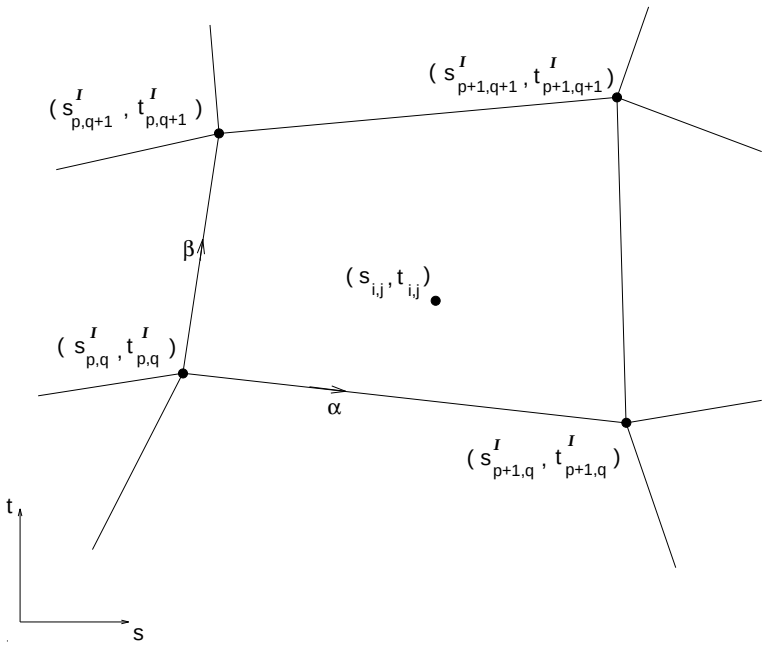

Figure 4: Patch (p,q) in parameter space $\mathcal{P}_{st}$

replaced by u^I_{ij} and v^I_{ij}. Compute the corresponding initial grid $\vec{x}^I_{ij}$ on surface $\mathcal{S}$ by $\vec{x}^I_{ij} = \vec{x}(\vec{u}^I_{ij})$.

step 3 Compute the normalized arclength parameters s and t at the boundary points of surface $\mathcal{S}$ in a way as described in Section 2.3 . Solve the two Laplace-Beltrami equations $\triangle s = 0, \triangle t = 0$, together with Dirichlet boundary conditions, on the initial grid $\vec{x}^I_{ij}$. The same solution procedure as described at the end of Section 2.4 can be used to discretize and solve the Laplace-Beltrami equations. Note that the Laplace-Beltrami equation $\triangle s = 0$ is in fact the same equation as defined by Eq.(50); the only difference compared to the two-dimensional case is that now $\vec{x} = (x, y, z)^T$ instead of $\vec{x} = (x, y)^T$. Thus in practice, the Laplace-Beltrami equations $\triangle s = 0$ and $\triangle t = 0$ are solved directly in computational space $\mathcal{C}$ instead of solving Eqs.(67),(68) on the nonuniform grid $\vec{u}^I_{ij}$ in parameter space $\mathcal{P}_{uv}$ which would be more complicated. Write the solution of the Laplace-Beltrami equations as $\{\vec{s}^I_{ij} = (s^I_{ij}, t^I_{ij}) \mid i = 0 \ldots N, j = 0 \ldots M\}$.

step 4 Compute in parameter space $\mathcal{P}_{st}$ the grid $\{\vec{s}_{ij} = (s_{ij}, t_{ij}) \mid i = 0 \ldots N, j = 0 \ldots M\}$ by applying the algebraic straight line algorithm according to Eqs.(39),(40).

step 5 Finally the inversion problem must be solved. Consider the parameter space $\mathcal{P}_{st}$ and consider the in **step 3** computed mesh (s^I_{ij}, t^I_{ij}) as an embedded nonuniform grid. This grid may also be considered as a non-overlapping subdivision of parameter space $\mathcal{P}_{st}$ by $N \times M$ patches where each patch has four corner points.

For a given interior grid point (i, j), the new position $\vec{x}_{ij}$ on surface $\mathcal{S}$ of the final grid is now obtained as follows. Locate the patch in parameter space $\mathcal{P}_{st}$ to which the in **step 4** computed value (s_{ij}, t_{ij}) belongs. Suppose that (s_{ij}, t_{ij}) belongs to patch (p,q) as shown in Fig.4.

The local patch parameters α and β are now defined by the following two bilinear equations

$$\begin{aligned} s_{i,j} &= s^I_{p,q}(1-\alpha)(1-\beta) + s^I_{p+1,q}\alpha(1-\beta) + s^I_{p,q+1}(1-\alpha)\beta + s^I_{p+1,q+1}\alpha\beta, \\ t_{i,j} &= t^I_{p,q}(1-\alpha)(1-\beta) + t^I_{p+1,q}\alpha(1-\beta) + t^I_{p,q+1}(1-\alpha)\beta + t^I_{p+1,q+1}\alpha\beta. \end{aligned}$$

The two parameters α and β are solved by Newton iteration. Note that $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$ because (s_{ij}, t_{ij}) belongs to patch (p,q). Compute the corresponding position $\vec{u}_{ij} = (u_{ij}, v_{ij})$ in parameter space $\mathcal{P}_{uv}$ by

$$\begin{aligned} u_{i,j} &= u^I_{p,q}(1-\alpha)(1-\beta) + u^I_{p+1,q}\alpha(1-\beta) + u^I_{p,q+1}(1-\alpha)\beta + u^I_{p+1,q+1}\alpha\beta, \\ v_{i,j} &= v^I_{p,q}(1-\alpha)(1-\beta) + v^I_{p+1,q}\alpha(1-\beta) + v^I_{p,q+1}(1-\alpha)\beta + v^I_{p+1,q+1}\alpha\beta, \end{aligned}$$

and compute $\vec{x}_{ij} = \vec{x}(\vec{u}_{ij})$ where $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$ is the given parametrization. The grid $\{\vec{x}_{ij} \mid i = 0 \ldots N, j = 0 \ldots M\}$ is the final surface grid.

The same algorithm can also be used to obtain an orthogonal grid at the boundary of surface $\mathcal{S}$. The only changes that have to be made are in **step 3** and **step 4**. In **step 3**, the Laplace-Beltrami equations must then be solved together with the Neumann boundary condition $\frac{\partial s}{\partial n} = 0$ along edges E_3 and E_4, and $\frac{\partial t}{\partial n} = 0$ along edges E_1 and E_2, where n is the outward normal direction. In **step 4**, the grid $\vec{s}_{ij}$ must be computed using Eqs.(47),(48) instead of Eqs.(5),(6).

4.3 Illustrations

We only consider parametrized surfaces which are defined as interpolated surfaces, constructed from a two-dimensional array of control points, and passing through these control points. The surface shape of each patch, spanned between four adjacent control points, is defined by a bi-cubic polynomial. Hermite interpolation is used to connect the surface shapes of the patches smoothly. For details, see [12] or Appendix B in [16]. The parametrization of a surface, defined by the mapping $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$, is constructed such that this mapping is continuously differentiable. The parametrization depends on the position of the control points.

As an illustration, consider a surface $\mathcal{S}$ which is defined by an irregular control point mesh in a unit square as shown in Fig.27. Thus the shape of surface $\mathcal{S}$ is a unit square. Fig.28 shows how a uniform grid in $\mathcal{P}_{uv}$ is mapped onto surface $\mathcal{S}$ by the parametrization $\vec{x} : \mathcal{P}_{uv} \mapsto \mathcal{S}$. This figure clearly demonstrates that the parametrization of $\mathcal{S}$ depends on the position of the control points.

The grid in Fig.28 is also the initial surface grid $\vec{x}^I_{ij}$ as defined at **step 2** of the grid generation algorithm described above. The corresponding initial grid $\vec{u}^I_{ij}$, also defined at **step 2**, is a uniform grid in parameter space $\mathcal{P}_{uv}$. The new grid $\vec{u}_{ij}$, defined at **step 5**, is shown in Fig.29. Note that the behaviour of the grid in Fig.29. is opposite to the behaviour of the grid in Fig.28. The corresponding final surface grid $\vec{x}_{ij}$, also defined at **step 5**, is shown in Fig.30. As expected, this surface grid is uniform. Thus this example clearly demonstrates that the surface grid is independent of the parametrization.

Another illustration of the fact that an elliptic surface grid is independent of the parametrization of the surface is shown in Figs.31,32. Fig.31 shows an irregularly distributed control point mesh on a smooth surface. The surface is defined as $z = \frac{1}{8}\tanh(15(\frac{1}{4} - (x-1)^2 - (y-1)^2))$,

$(x, y) \in [0, 1]^2$. Fig.32 shows an elliptic grid. Equidistributed boundary grid points are used as Dirichlet boundary condition. This figure clearly demonstrates that the interior surface grid only depends on the shape of the surface and is independent of the parametrization.

A more practical example is shown in Fig.33 where the control point mesh is shown of a surface $\mathcal{S}$ which belongs to the lower part of a wing near the intersection of a pylon. Fig.34 is a close-up of the control point distribution near the leading edge of the pylon. Fig.35 shows a detail of the initial surface grid $\vec{x}^I_{ij}$ as defined at **step 2**. This grid is badly distributed and shows the influence of the control point mesh. Fig.36 shows a detail of the final elliptic surface grid $\vec{x}_{ij}$ as defined at **step 5**. The elliptic surface grid shows no dependency of the control point mesh and has a good interior grid point distribution.

5 3D Grid Generation

5.1 Derivation of the 3D grid generation equations

The two dimensional grid generation method described in Section 2 can be extended into three dimensions.

Consider a simply connected bounded domain $\mathcal{D}$ in three dimensional space with Cartesian coordinates $\vec{x} = (x, y, z)^T$. Suppose that $\mathcal{D}$ is bounded by six faces $F_1, F_2, F_3, F_4, F_5, F_6$. Let (F_1, F_2) , (F_3, F_4) and (F_5, F_6) be the three pairs of opposite faces. Furthermore, consider the twelve edges $\{E_i, i = 1 \ldots 12\}$ and assume that these edges are related to the six faces as shown in Fig.5

Consider the computational space $\mathcal{C}$ as the unit cube in three dimensional space with Cartesian coordinates $\vec{\xi} = (\xi, \eta, \zeta)^T$. Assume that a mapping $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed which maps the boundary of $\mathcal{C}$ one-to-one on the boundary of $\mathcal{D}$. This mapping defines the boundary grid point distribution. Assume that

- $\xi \equiv 0$ at face F_1 and $\xi \equiv 1$ at face F_2,
- $\eta \equiv 0$ at face F_3 and $\eta \equiv 1$ at face F_4,
- $\zeta \equiv 0$ at face F_5 and $\zeta \equiv 1$ at face F_6.

We wish to construct a mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ which obeys the boundary conditions and which is a differentiable one-to-one mapping. Furthermore, we require that the interior grid point distribution is a good reflection of the prescribed boundary grid point distribution.

As in two dimensions, this mapping will be a composition of an algebraic transformation and an elliptic transformation based on the Laplace equations. The algebraic transformation is a differentiable one-to-one mapping from computational space onto a parameter space $\mathcal{P}$. The parameter space $\mathcal{P}$ is also a unit cube. The elliptic transformation is a differentiable one-to-one mapping from parameter space to domain $\mathcal{D}$.

Introduce the parameter space $\mathcal{P}$ as the unit cube in three dimensional space with Cartesian coordinates $\vec{s} = (s, t, u)^T$. Require that the parameters s, t and u obey:

- $s \equiv 0$ at face F_1 and $s \equiv 1$ at face F_2,
- $t \equiv 0$ at face F_3 and $t \equiv 1$ at face F_4,
- $u \equiv 0$ at face F_5 and $u \equiv 1$ at face F_6,

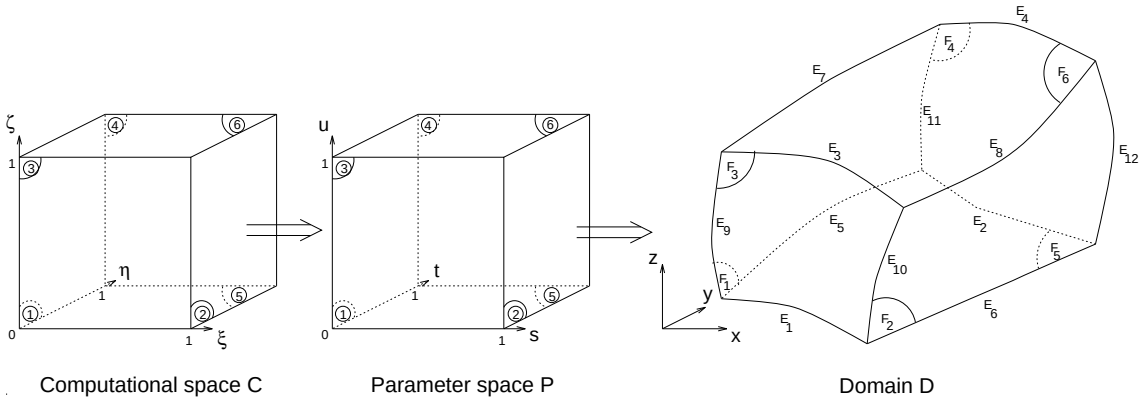

Figure 5: Transformation from computational (ξ, η, ζ) space to a domain $\mathcal{D}$ in Cartesian (x, y, z) space.

- s is the normalized arclength at edges E_1, E_2, E_3, E_4,
- t is the normalized arclength at edges E_5, E_6, E_7, E_8,
- u is the normalized arclength at edges $E_9, E_{10}, E_{11}, E_{12}$.

From the first three requirements it follows that

- $s(0, \eta, \zeta) = 0$ and $s(1, \eta, \zeta) = 1$,
- $t(\xi, 0, \zeta) = 0$ and $t(\xi, 1, \zeta) = 1$,
- $u(\xi, \eta, 0) = 0$ and $u(\xi, \eta, 1) = 1$.

The coordinates (s, t, u) are defined at all twelve edges of domain $\mathcal{D}$. The computational coordinates are defined at the complete boundary of $\mathcal{D}$ and thus also at the twelve edges. Thus each point at the twelve edges of domain $\mathcal{D}$ has a unique (ξ, η, ζ) coordinate and a unique (s, t, u) coordinate. Thus each (ξ, η, ζ) value at the twelve edges of the unit cube in computational space has also a unique (s, t, u) value. Hence, we may conclude that:

- $s(\xi, 0, 0) = s_{E_1}(\xi), s(\xi, 1, 0) = s_{E_2}(\xi), s(\xi, 0, 1) = s_{E_3}(\xi), s(\xi, 1, 1) = s_{E_4}(\xi)$,
- $t(0, \eta, 0) = t_{E_5}(\eta), t(1, \eta, 0) = t_{E_6}(\eta), t(0, \eta, 1) = t_{E_7}(\eta), t(1, \eta, 1) = t_{E_8}(\eta)$,
- $u(0, 0, \zeta) = u_{E_9}(\zeta), u(1, 0, \zeta) = u_{E_{10}}(\zeta), u(0, 1, \zeta) = u_{E_{11}}(\zeta), u(1, 1, \zeta) = u_{E_{12}}(\zeta)$.

The twelve edge functions $s_{E_1}, \ldots, u_{E_{12}}$ are monotonically increasing and are defined by the prescribed boundary point distribution at the twelve edges.

The algebraic mapping from computational space to parameter space, $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$, is now defined as

$$s = s_{E_1}(\xi)(1-t)(1-u) + s_{E_2}(\xi)t(1-u) + s_{E_3}(\xi)(1-t)u + s_{E_4}(\xi)tu, \quad (70)$$

$$t = t_{E_5}(\eta)(1-s)(1-u) + t_{E_6}(\eta)s(1-u) + t_{E_7}(\eta)(1-s)u + t_{E_8}(\eta)su, \quad (71)$$

$$u = u_{E_9}(\zeta)(1-s)(1-t) + u_{E_{10}}(\zeta)s(1-t) + u_{E_{11}}(\zeta)(1-s)t + u_{E_{12}}(\zeta)st. \quad (72)$$

Note that this mapping only depends on the boundary grid point distribution at the twelve edges of domain $\mathcal{D}$.

Eq.(70) implies that a grid plane ξ = constant is mapped to the parameter space $\mathcal{P}$ as a bilinear surface: s is a bilinear function of t and u. Similarly, Eq.(71) and Eq.(72) imply that grid planes η = constant and ζ = constant are also mapped to the parameter space $\mathcal{P}$ as bilinear surfaces. For a given computational coordinate (ξ, η, ζ) the corresponding (s, t, u) value is found as the intersection point of three bilinear surfaces. For this reason, the system defined by Eqs.(70),(71),(72) is called the "algebraic bilinear transformation" because of the use of bilinear surfaces in parameter space $\mathcal{P}$. The algebraic bilinear transformation is the three dimensional equivalent of the two dimensional algebraic straight line transformation. It can be easily verified that two bilinear surfaces corresponding to two different ξ-values will never intersect in parameter space $\mathcal{P}$. The same is true for two different η or ζ values. This observation indicates that the algebraic transformation is a differentiable one-to-one mapping. The system defined by Eqs.(70),(71),(72) can also be interpreted as a transfinite interpolation with nonlinear blending functions.

Because $\vec{x} : \partial\mathcal{C} \mapsto \partial\mathcal{D}$ is prescribed and $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ is defined by the algebraic bilinear transformation, it follows that the (s, t, u) coordinates are now defined at the complete boundary of domain $\mathcal{D}$ including the interior of the six faces $F_1, \ldots, F_6$.

Require that (s, t, u) are harmonic functions in the interior of $\mathcal{D}$, i.e.

$$\triangle s = s_{xx} + s_{yy} + s_{zz} = 0 \ , \ \triangle t = t_{xx} + t_{yy} + t_{zz} = 0 \ , \ \triangle u = u_{xx} + u_{yy} + u_{zz} = 0 \ . \qquad (73)$$

Thus a linear elliptic boundary value problem defines the mapping $\vec{s} : \mathcal{D} \mapsto \mathcal{P}$. It seems to be still an open theoretical question whether this mapping is one-to-one [5]. The proof in [4] is not correct. However, in this paper it is assumed that $\vec{s} : \mathcal{D} \mapsto \mathcal{P}$ is one-to-one and thus that the inverse mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ exists. This inverse mapping obeys a nonlinear system of elliptic differential equations.

Note that the mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ is not independent of the boundary grid point distribution and may thus not be considered as a property of domain $\mathcal{D}$. This is because the (s, t, u) coordinates at the interior of the six boundary faces depend on the boundary grid point distribution. It is possible to define the mapping $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ independently from the boundary grid point distribution by requiring that the (s, t, u) coordinates obey the Laplace-Beltrami equations in the interior of the six faces of domain $\mathcal{D}$ but then it is no longer possible to use the simple algebraic bilinear transformation defined by Eqs.(70),(71),(72).

The algebraic transformation $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$ and the elliptic transformation $\vec{x} : \mathcal{P} \mapsto \mathcal{D}$ are thus assumed to be differentiable one-to-one mappings. Then the composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$, defined as $\vec{x} = \vec{x}(\vec{s}(\vec{\xi}))$, is also differentiable and one-to-one. Furthermore, due to the properties of the basic mappings, we may indeed expect that the interior grid point distribution will be a good reflection of the boundary point distribution.

The composite mapping $\vec{x} : \mathcal{C} \mapsto \mathcal{D}$ obeys an elliptic Poisson system with control functions defined by the algebraic mapping $\vec{s} : \mathcal{C} \mapsto \mathcal{P}$. This three dimensional elliptic Poisson system together with the appropriate expressions of the control functions is a straightforward extension of the two dimensional system and will be derived in the remainder of this section.

Define the three covariant base vectors

$$\vec{a}_1 = \vec{x}_\xi \ , \ \vec{a}_2 = \vec{x}_\eta \ , \ \vec{a}_3 = \vec{x}_\zeta . \qquad (74)$$

and the covariant metric tensor components

$$a_{ij} = (\vec{a}_i, \vec{a}_j) \ , \ i = \{1, 2, 3\}, j = \{1, 2, 3\}. \qquad (75)$$

The three contravariant base vectors $\vec{a}^1$, $\vec{a}^2$ and $\vec{a}^3$ are defined according to the rules

$$(\vec{a}^i, \vec{a}_j) = \delta^i_j \ , \ i = \{1,2,3\}, j = \{1,2,3\}. \tag{76}$$

The contravariant metric tensor components

$$a^{ij} = (\vec{a}^i, \vec{a}^j) \ , \ i = \{1,2,3\}, j = \{1,2,3\}, \tag{77}$$

fulfill

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \begin{pmatrix} a^{11} & a^{12} & a^{13} \\ a^{12} & a^{22} & a^{23} \\ a^{13} & a^{23} & a^{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{78}$$

The three contravariant base vectors can be expressed as

$$\vec{a}^1 = a^{11}\vec{a}_1 + a^{12}\vec{a}_2 + a^{13}\vec{a}_3 \ , \ \vec{a}^2 = a^{12}\vec{a}_1 + a^{22}\vec{a}_2 + a^{23}\vec{a}_3 \ , \ \vec{a}^3 = a^{13}\vec{a}_1 + a^{23}\vec{a}_2 + a^{33}\vec{a}_3. \tag{79}$$

Define J^2 as the determinant of the covariant metric tensor.

Consider an arbitrary function $\phi = \phi(\xi, \eta, \zeta)$. Then ϕ is also defined in domain $\mathcal{D}$ and the Laplacian of ϕ can be expressed as

$$\begin{aligned} \triangle\phi = \frac{1}{J}\Big\{ &\left(Ja^{11}\phi_\xi + Ja^{12}\phi_\eta + Ja^{13}\phi_\zeta\right)_\xi + \left(Ja^{12}\phi_\xi + Ja^{22}\phi_\eta + Ja^{23}\phi_\zeta\right)_\eta \\ &+ \left(Ja^{13}\phi_\xi + Ja^{23}\phi_\eta + Ja^{33}\phi_\zeta\right)_\zeta \Big\}. \end{aligned} \tag{80}$$

As in the two-dimensional case, substitution of $\phi \equiv \xi$, $\phi \equiv \eta$ and $\phi \equiv \zeta$ into this equation yields expressions for $\triangle\xi$, $\triangle\eta$ and $\triangle\zeta$. Combining these expressions with Eq.(80) gives

$$\triangle\phi = a^{11}\phi_{\xi\xi} + 2a^{12}\phi_{\xi\eta} + 2a^{13}\phi_{\xi\zeta} + a^{22}\phi_{\eta\eta} + 2a^{23}\phi_{\eta\zeta} + a^{33}\phi_{\zeta\zeta} + \triangle\xi\phi_\xi + \triangle\eta\phi_\eta + \triangle\zeta\phi_\zeta \tag{81}$$

Substitute $\phi = (s,t,u)^T$ in Eq.(81) and use the property that s, t and u are harmonic in domain $\mathcal{D}$, i.e. $\triangle s = 0$, $\triangle t = 0$ and $\triangle u = 0$. Then the following expressions for the Laplacian of ξ , η and ζ are found:

$$\begin{pmatrix} \triangle\xi \\ \triangle\eta \\ \triangle\zeta \end{pmatrix} = a^{11}\vec{P}_{11} + 2a^{12}\vec{P}_{12} + 2a^{13}\vec{P}_{13} + a^{22}\vec{P}_{22} + 2a^{23}\vec{P}_{23} + a^{33}\vec{P}_{33}, \tag{82}$$

where

$$\begin{aligned} \vec{P}_{11} &= -T^{-1}\begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \\ u_{\xi\xi} \end{pmatrix} \ , \ \vec{P}_{12} = -T^{-1}\begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \\ u_{\xi\eta} \end{pmatrix} \ , \ \vec{P}_{13} = -T^{-1}\begin{pmatrix} s_{\xi\zeta} \\ t_{\xi\zeta} \\ u_{\xi\zeta} \end{pmatrix} \ , \\ \vec{P}_{22} &= -T^{-1}\begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \\ u_{\eta\eta} \end{pmatrix} \ , \ \vec{P}_{23} = -T^{-1}\begin{pmatrix} s_{\eta\zeta} \\ t_{\eta\zeta} \\ u_{\eta\zeta} \end{pmatrix} \ , \ \vec{P}_{33} = -T^{-1}\begin{pmatrix} s_{\zeta\zeta} \\ t_{\zeta\zeta} \\ u_{\zeta\zeta} \end{pmatrix}, \end{aligned} \tag{83}$$

and the matrix T is defined as

$$T = \begin{pmatrix} s_\xi & s_\eta & s_\zeta \\ t_\xi & t_\eta & t_\zeta \\ u_\xi & u_\eta & u_\zeta \end{pmatrix}. \tag{84}$$

The 18 coefficients of the six vectors $\vec{P}_{11}$,$\vec{P}_{12}$,$\vec{P}_{13}$, $\vec{P}_{22}$,$\vec{P}_{23}$,$\vec{P}_{33}$ are so called control functions. Thus the 18 control functions are completely defined and easily computed for a given algebraic transformation mapping $\vec{s} = \vec{s}(\vec{\xi})$. Different and less useful expressions of these control functions can also be found in [1, 2].

Finally, substitution of $\phi \equiv \vec{x}$ in Eq.(81) and using the fact that $\triangle \vec{x} \equiv 0$ we arrive at the following equation

$$a^{11}\vec{x}_{\xi\xi} + 2a^{12}\vec{x}_{\xi\eta} + 2a^{13}\vec{x}_{\xi\zeta} + a^{22}\vec{x}_{\eta\eta} + 2a^{23}\vec{x}_{\eta\zeta} + a^{33}\vec{x}_{\zeta\zeta} + \triangle\xi\vec{x}_{\xi} + \triangle\eta\vec{x}_{\eta} + \triangle\zeta\vec{x}_{\zeta} = 0. \quad (85)$$

The final form of the Poisson grid generation system can now be derived from this equation by substitution of Eq.(82),by multiplication with J^2, and by expressing the contravariant tensor components in the covariant tensor components according to Eq.(78). The result can be written as:

$$\begin{aligned} & \alpha^{11}\vec{x}_{\xi\xi} + 2\alpha^{12}\vec{x}_{\xi\eta} + 2\alpha^{13}\vec{x}_{\xi\zeta} + \alpha^{22}\vec{x}_{\eta\eta} + 2\alpha^{23}\vec{x}_{\eta\zeta} + \alpha^{33}\vec{x}_{\zeta\zeta} \\ + & \left(\alpha^{11}P^1_{11} + 2\alpha^{12}P^1_{12} + 2\alpha^{13}P^1_{13} + \alpha^{22}P^1_{22} + 2\alpha^{23}P^1_{23} + \alpha^{33}P^1_{33}\right)\vec{x}_{\xi} \\ + & \left(\alpha^{11}P^2_{11} + 2\alpha^{12}P^2_{12} + 2\alpha^{13}P^2_{13} + \alpha^{22}P^2_{22} + 2\alpha^{23}P^2_{23} + \alpha^{33}P^2_{33}\right)\vec{x}_{\eta} \\ + & \left(\alpha^{11}P^3_{11} + 2\alpha^{12}P^3_{12} + 2\alpha^{13}P^3_{13} + \alpha^{22}P^3_{22} + 2\alpha^{23}P^3_{23} + \alpha^{33}P^3_{33}\right)\vec{x}_{\zeta} = 0, \end{aligned} \quad (86)$$

with

$$\begin{aligned} \alpha^{11} &= a_{22}a_{33} - a_{23}^2\ ,\ \alpha^{12} = a_{13}a_{23} - a_{12}a_{33}\ ,\ \alpha^{13} = a_{12}a_{23} - a_{13}a_{22}\ , \\ \alpha^{22} &= a_{11}a_{33} - a_{13}^2\ ,\ \alpha^{23} = a_{13}a_{12} - a_{11}a_{23}\ ,\ \alpha^{33} = a_{11}a_{22} - a_{12}^2, \end{aligned} \quad (87)$$

and

$$\begin{aligned} a_{11} &= (\vec{x}_{\xi}, \vec{x}_{\xi})\ ,\ a_{12} = (\vec{x}_{\xi}, \vec{x}_{\eta})\ ,\ a_{13} = (\vec{x}_{\xi}, \vec{x}_{\zeta})\ , \\ a_{22} &= (\vec{x}_{\eta}, \vec{x}_{\eta})\ ,\ a_{23} = (\vec{x}_{\eta}, \vec{x}_{\zeta})\ ,\ a_{33} = (\vec{x}_{\zeta}, \vec{x}_{\zeta})\ . \end{aligned} \quad (88)$$

Eq.(86), together with the expressions for the control functions P^k_{ij} given by Eq.(83), forms our 3D grid generation system. Grids are computed by solving this quasi-linear system of elliptic partial differential equations with the prescribed boundary grid points as Dirichlet boundary conditions. The discretization and solution of this Poisson system is described in the next section.

5.2 Discretization and solution method

Consider a rectangular grid of $(N+1) \times (M+1) \times (L+1)$ points in computational space $\mathcal{C}$ defined as

$$\begin{aligned} \xi_{i,j,k} = \xi_i = i/N\ ,\ \eta_{i,j,k} = \eta_j = j/M\ ,\ \zeta_{i,j,k} = \zeta_k = k/L\ ,\ i &= 0 \ldots N,\ j = 0 \ldots M, \\ k &= 0 \ldots L. \end{aligned} \quad (89)$$

Assume that $\vec{x}_{i,j,k}$ is prescribed on the boundary of this grid and consider the computation of $\vec{x}_{i,j,k}$ in the interior of the computational grid based on the solution of the Poisson system defined by Eq.(86).

The first task is the computation of the algebraic transformation. The computation of the arclength normalized values at the twelve edges is straightforward and performed

in exactly the same way as described in Section 2.3 . The arclength normalized variables $(s_{i,j,k}, t_{i,j,k}, u_{i,j,k})$ in the interior of the six boundary faces and in the interior of the grid are computed according to the algebraic bilinear transformation defined by Eqs.(70),(71),(72) and are thus found by solving simultaneously the three bilinear algebraic equations:

$$\begin{aligned} s_{i,j,k} = s_{i,0,0}(1-t_{i,j,k})(1-u_{i,j,k}) &+ s_{i,M,0}t_{i,j,k}(1-u_{i,j,k}) \\ &+ s_{i,0,L}(1-t_{i,j,k})u_{i,j,k} + s_{i,M,L}t_{i,j,k}u_{i,j,k}, \qquad (90) \\ t_{i,j,k} = t_{0,j,0}(1-s_{i,j,k})(1-u_{i,j,k}) &+ t_{N,j,0}s_{i,j,k}(1-u_{i,j,k}) \\ &+ t_{0,j,L}(1-s_{i,j,k})u_{i,j,k} + t_{N,j,L}s_{i,j,k}u_{i,j,k}, \qquad (91) \\ u_{i,j,k} = u_{0,0,k}(1-s_{i,j,k})(1-t_{i,j,k}) &+ u_{N,0,k}s_{i,j,k}(1-t_{i,j,k}) \\ &+ u_{0,M,k}(1-s_{i,j,k})t_{i,j,k} + u_{N,M,k}s_{i,j,k}t_{i,j,k}. \qquad (92) \end{aligned}$$

for each pair $(i,j,k) \in (0\ldots N, 0\ldots M, 0\ldots L)$.

At each grid point (i,j,k), the 18 control functions defined by Eq.(83), are now easily computed using central difference representations of the derivatives of s, t and u.

What remains is the iterative solution process of the nonlinear elliptic Poisson grid generation system defined by Eq.(86). The discretization and the applied Picard iteration process is similar as used to solve the 2D Poisson grid generation systems and details are therefore omitted.

During a Picard iteration, a linear system of equations must be solved for the unknowns $\vec{x}_{i,j,k}, i = 0\ldots N, j = 0\ldots M, k = 0\ldots L$. This linear system consists of 19-point stencils with Dirichlet boundary conditions. Fig.6 shows the structure of the 19-point stencils. Such linear systems are solved by another black-box linear-system solver, also developed at C.W.I. by P.M. de Zeeuw. The black-box linear-system solver, called THREED, is based on multigrid and Bi-CGSTAB [18, 19]. The linear system solver THREED is called three times to compute the three components $x_{i,j,k}$,$y_{i,j,k}$ and $z_{i,j,k}$ of $\vec{x}_{i,j,k}$.

The complete process is repeated until a sufficiently accurate solution has been obtained. The initial start solution $\vec{x}^0$ is obtained by algebraic grid generation. The final grid is independent of the initial grid. Moreover, the quality of the initial grid is unimportant and severe grid folding of the initial grid is allowed. In general, about 10 Picard iterations are enough to obtain a sufficiently accurate solution of the nonlinear elliptic Poisson equations.

5.3 Illustrations

An example of a multi-block Navier-Stokes is shown in Figs.37,...,42. The mesh is a 4 blocks CO-type grid around an Onera-M6 wing. The total number of grid cells is $256 \times 64 \times 48$. On the wing, the mesh-width in normal direction of the first grid cell is at the leading edge 10^{-5} times the local chord-length and at the trailing edge 2×10^{-5} times the local chord-length .

Fig.37 and Fig.38 are three-dimensional views of vertical grid-planes intersecting the wing. Fig.39 is a close-up of the wing-tip. Fig.40 is a two-dimensional projection of horizontal grid-planes intersecting the wing. Fig.41 and Fig.42 are two-dimensional projections of vertical grid-planes at stations halfway in spanwise and chordwise direction.

The grid is grid-folding free and the interior grid point distribution is a good reflection of the prescribed boundary grid point distribution at the block-faces.

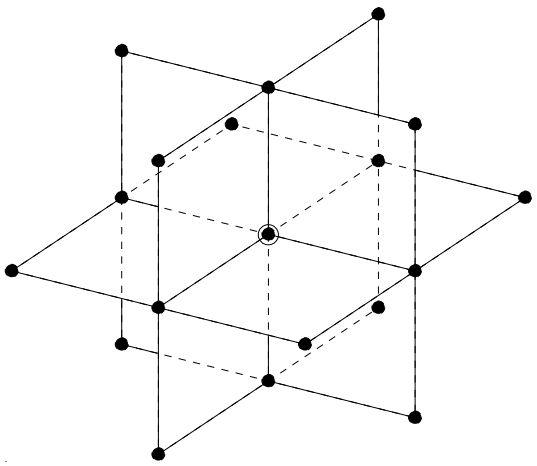

Figure 6: Structure of 19-point stencil of the discretized 3D Poisson grid generation system.

6 Conclusions

An elliptic grid generation method is developed which produce excellent grids in the sense of smoothness, grid point distribution and regularity. The elliptic grid generation method is based on the composition of an algebraic and elliptic transformation. The elliptic transformation is based on the Laplace equations for domains, and on the Laplace-Beltrami equations for surfaces. The composite mappings obey the familiar grid generation systems of Poisson equations with control functions specified by the algebraic transformation. New expressions for the control functions are derived which only depend on the algebraic transformation and not also on the inverse of this transformation. The composite mappings are differentiable, and surely one-to-one for 2D domains and surfaces, and in practice also for 3D domains.

It is described how the proposed elliptic grid generation method can be used to generate boundary conforming grids in 2D domains, 3D domains and surfaces. It is shown that surface grid generation on minimal surfaces (soap films) is in fact a straightforward extension of grid generation in 2D domains. It is also shown that grid generation on parametrized surfaces with a prescribed geometrical shape can be performed very easily by only solving two linear elliptic partial differential equations and an inversion problem. A generated surface grid on a parametrized surface is independent of the parametrization itself and only depends on the shape of the surface and the prescribed boundary grid point distribution.

For 2D domains and surfaces, it is described how the algebraic transformation can be redefined to obtain grids which are orthogonal at the boundary.

The described elliptic grid generation method has been implemented into NLR's multi-block grid generation code ENGRID and is extensively used for the generation of boundary conforming Navier-Stokes grids in blocks and block-faces with very complex aerodynamic shapes, like complete aircraft configurations with propulsion systems and tailplanes.

References

[1] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, Numerical Grid Generation: Foundations and Applications, Elsevier, New York, 1985.

[2] Z.U.A. Warsi, Basic Differential Models for Coordinate Generation, in: Proceedings Numerical Grid Generation, J.F. Thompson (ed), pp. 41-77, North-Holland,Amsterdam,1982.

[3] Z.U.A. Warsi, Numerical Grid Generation in Arbitrary Surfaces through a Second-Order Differential-Geometric Model, J. Comput. Phys. 64, pp.82-96, 1986.

[4] C. W. Mastin and J. F. Thompson, Transformation of Three-Dimensional Regions onto Rectangular Regions by Elliptic Systems, Numer. Math. 29,pp. 397-407, 1978.

[5] G. Liao, On Harmonic Maps, in: Frontiers in Applied Mathematics 8, Mathematical Aspects of Numerical Grid Generation, J.E. Castillo (ed), pp.123-130, SIAM,Philadelphia,1991.

[6] B.K. Soni, Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics, in: Proceedings, AIAA 7th Computational Fluid Dynamics Conference, pp. 351-359, AIAA paper 85-1526,1985.

[7] P. J. Roache, S. Steinberg, A new Approach to Grid Generation Using a Variational Formulation, in: Proceedings, AIAA 7th Computational Fluid Dynamics Conference, pp. 360-370, AIAA paper 85-1527,1985.

[8] P.D. Thomas, J.F. Middlecoff, Direct Control of Grid Point Distribution in Meshes Generated by Elliptic Equations, in: AIAA journal 18, p. 652, 1980.

[9] R.L. Sorenson, Three-Dimensional Zonal Grids about Arbitrary Shapes by Poisson's Equation, in: Proceedings, Numerical Grid Generation in Computational Fluid Mechanics '88,S. Sengupta et al (ed), pp.75-84, Pineridge, Swansea, 1988.

[10] T. Sonar, Grid Generation Using Elliptic Partial Differential Equations, DFVLR-Forschungsbericht 89-15,1989.

[11] J. W. Boerstoel, J. M. J. W. Jacobs, A. Kassies, A. Amendola, R. Tognaccini and P. L. Vitagliano, Design and Testing of a Multi-Block Grid-Generation Procedure for Aircraft Design and Research, in: Proceedings, Applications of Mesh Generation to Complex 3-D Configurations, Agard CP 464,1990.

[12] J.W. Boerstoel, S.P. Spekreijse, P.L. Vitagliano, The Design of a System of Codes for Industrial Calculations of Flows around Aircraft and other Complex Aerodynamic Configurations, NLR-TP-92190, AIAA paper 92-2619, 10th Applied Aerodynamics Conference, Palo Alto, 1992.

[13] S.P. Spekreijse, J.W. Boerstoel, P.L. Vitagliano, J.L. Kuyvenhoven, Domain modeling and Grid Generation for Multi-block Structured Grids with Application to Aerodynamic and Hydrodynamic Configurations. In: Software Systems for Surface Modeling and Grid Generation, R.E. Smith (ed), NASA-CP-3143, pp.207-229,1992.

[14] S.P. Spekreijse, J.W. Boerstoel, J.L. Kuyvenhoven, M.J. van der Marel, Surface Grid Generation for Multi-block Structured Grids, In: Proceedings of the First European Computational Fluid Dynamics Conference, Eds. Ch. Hirsch, J. Periaux, W. Kordulla, Vol.2, p.937, Elsevier,1992.

[15] S.P. Spekreijse, J.W. Boerstoel, New Concepts for Multi-Block Grid Generation for Flow Domains Around Complex Aerodynamic Configurations. Numerical Grid Generation, In: Computational Fluid Dynamics and Related Fields, ed. A.S. Arcilla e.a., Elsevier North-Holland, 1991.

[16] S.P. Spekreijse, J.W. Boerstoel, Multiblock Grid Generation. Part 2 : Multiblock Aspects. This VKI-Lecture-Serie,1996.

[17] P.M. de Zeeuw, Matrix-dependent Prolongations and Restrictions in a Blackbox Multigrid Solver, Journal of Computational and Applied Mathematics,33,p.1,1990.

[18] P.M. de Zeeuw, Incomplete Line LU for Discretized Coupled PDEs as Preconditioner in Bi-CGSTAB, Report NM-R9213, Dept. of Numerical Mathematics, CWI, Amsterdam,1992.

[19] H.A. van der Vorst, Bi-CGSTAB: a Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, Siam J.Sci.Statis.Comput. 13(2),pp. 631-644, 1992.

[20] S.R. Chakravarthy, K.Y. Szema, U.C. Goldberg, J.J. Gorski, Application of a New Class of High Accuracy TVD Schemes to the Navier-Stokes Equations, AIAA paper 85-0165, AIAA 23rd Aerospace Science Meeting, Reno,Nevada,1985.

[21] E. Kreyszig, Introduction to Differential Geometry and Riemannian Geometry, Mathematical Expositions no.16, University of Toronto Press,1967.

[22] U.Dierkes, S. Hildebrandt, A. Kuster, O. Wohlrab, Minimal Surfaces I, Grundlehren der Mathematischen Wissenschaften 295, Springer Verlag, Berlin, 1991.

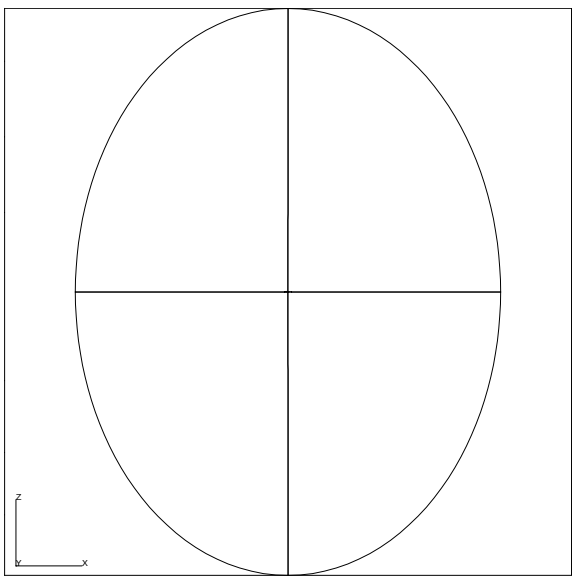

Figure 7: Region about NACA0012 airfoil subdivided into four domains.

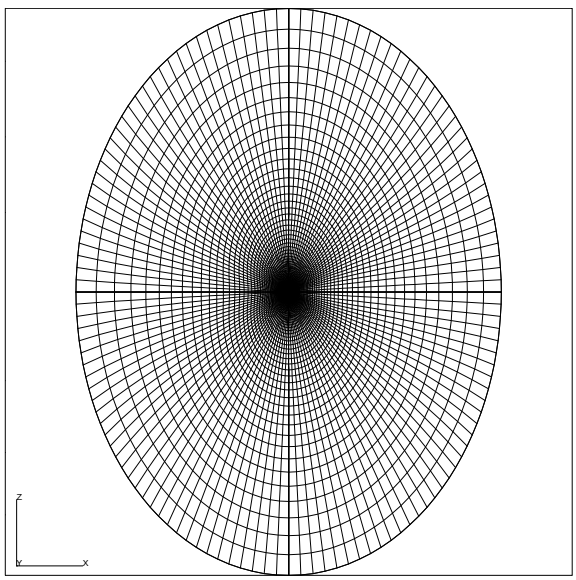

Figure 8: Complete O-type Euler grid.

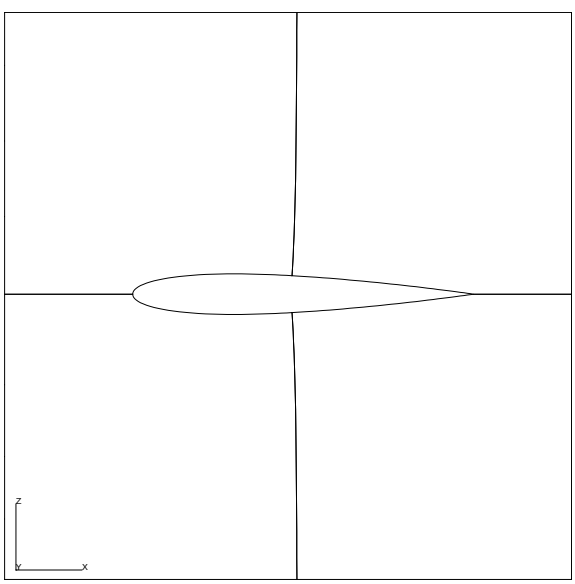

Figure 9: Domain boundaries near NACA0012 airfoil.

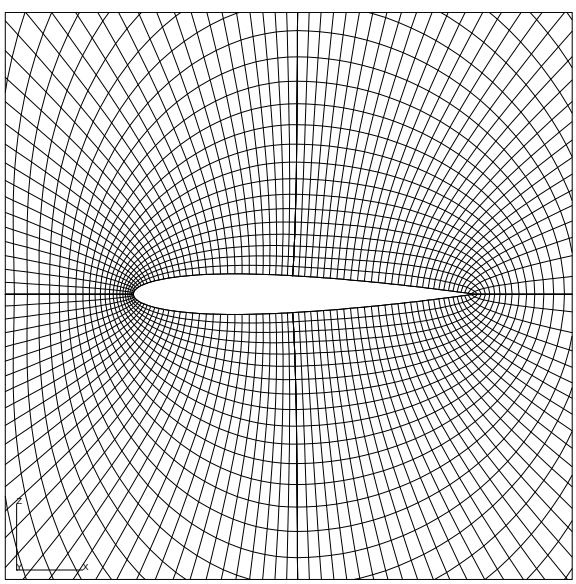

Figure 10: Grid near NACA0012 airfoil.

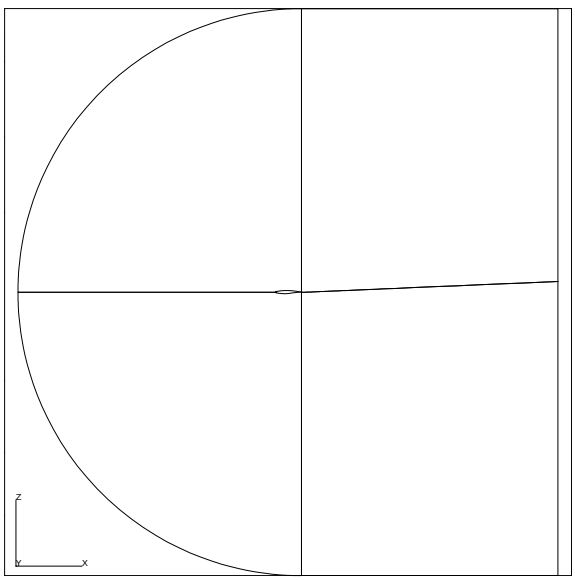

Figure 11: Region about RAE2822 airfoil subdivided into four domains.

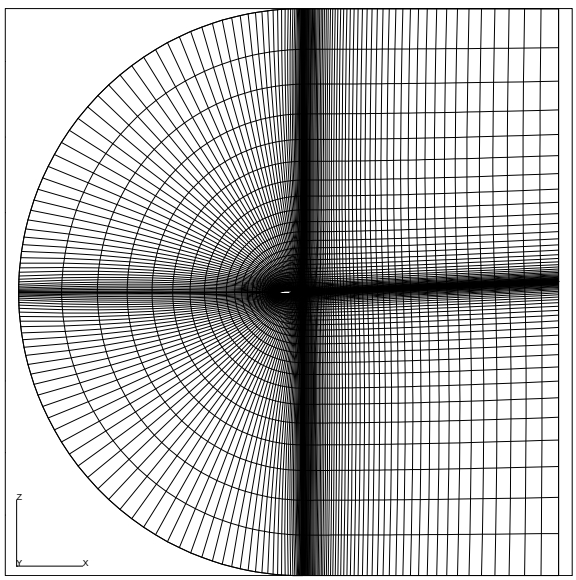

Figure 12: Complete C-type Navier-Stokes grid.

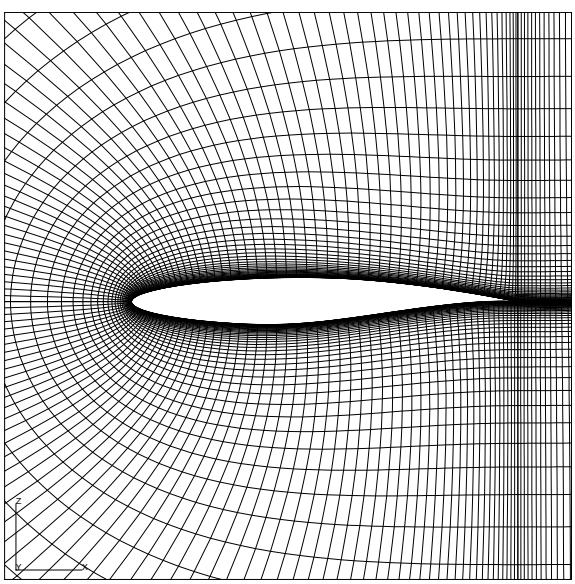

Figure 13: Grid near RAE2822 airfoil.

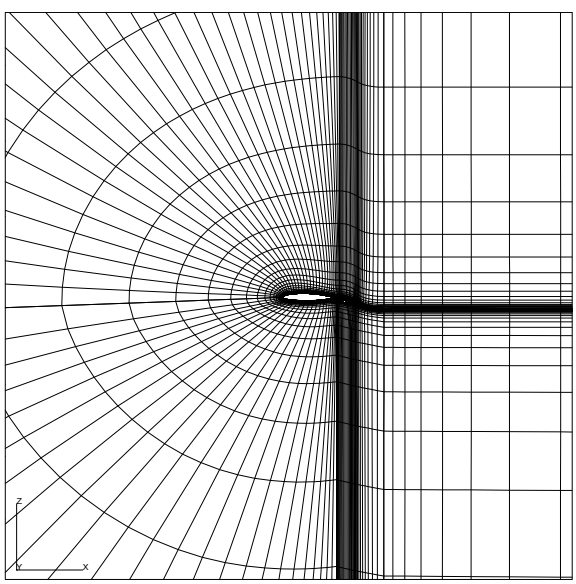

Figure 14: C-type Navier-Stokes grid about a wing with flap.

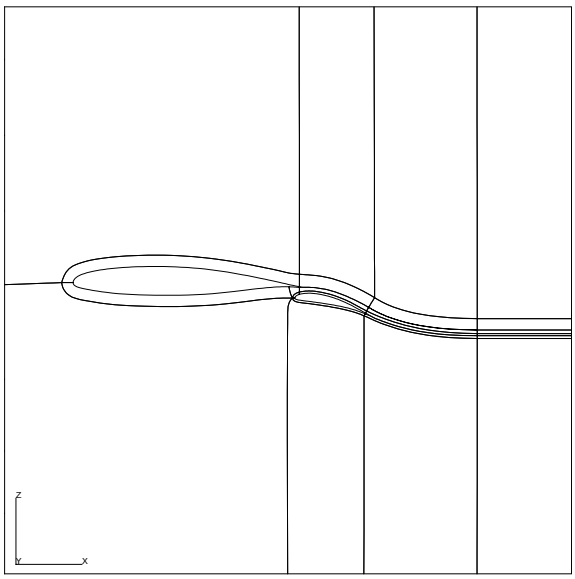

Figure 15: Domain decomposition about a wing with flap.

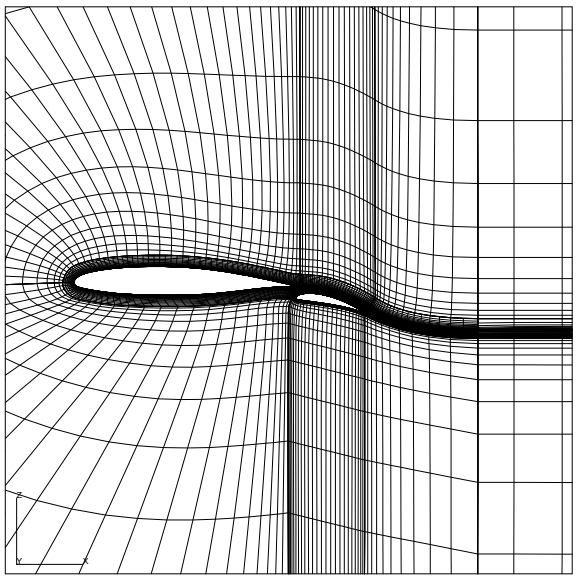

Figure 16: Corresponding multi-block grid.

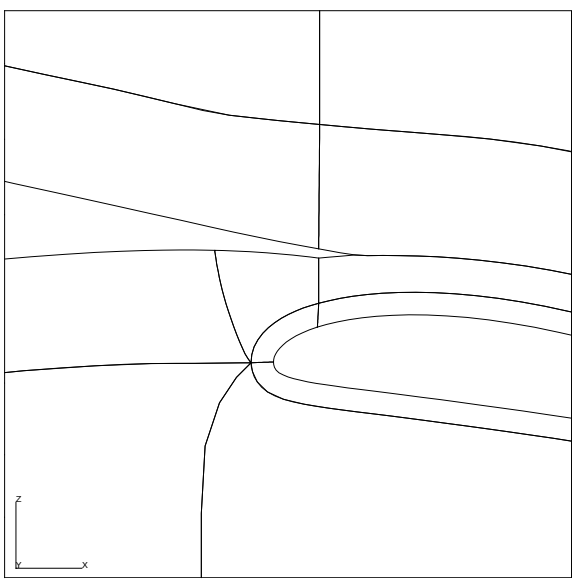

Figure 17: Blow-up of domain decomposition.

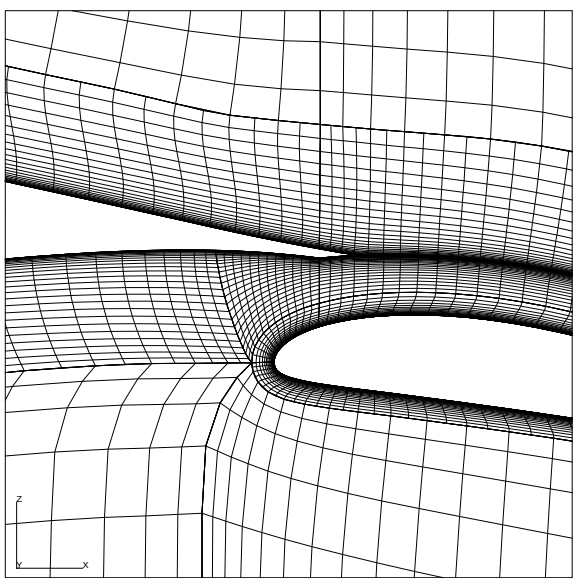

Figure 18: Blow-up of grid.

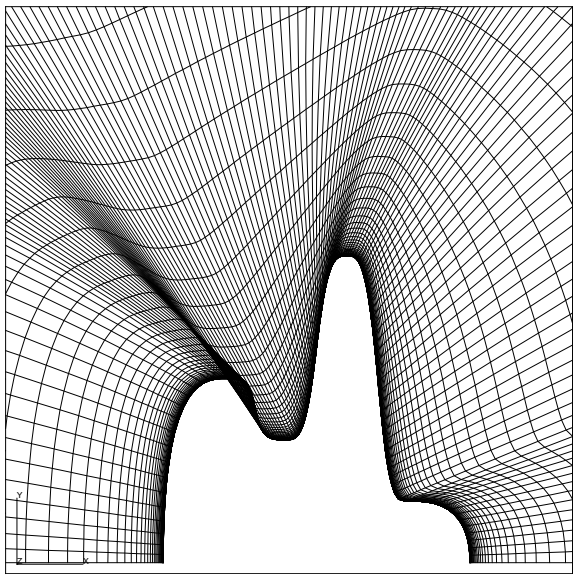

Figure 19: Initial algebraic grid with severe grid folding around a complex artificial boundary.

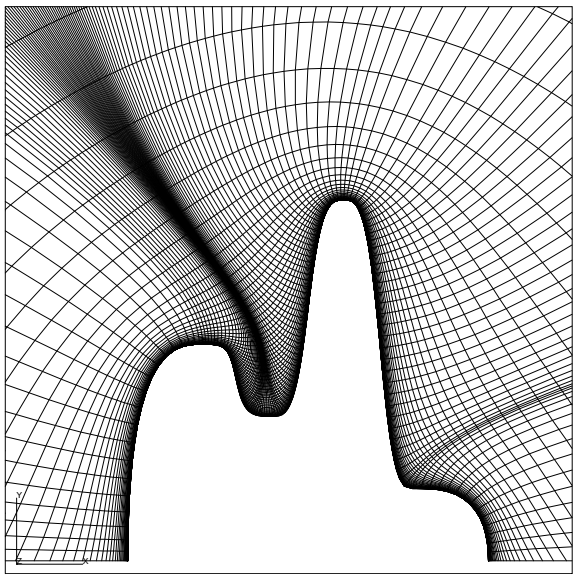

Figure 20: Elliptic grid with orthogonality at the boundary.

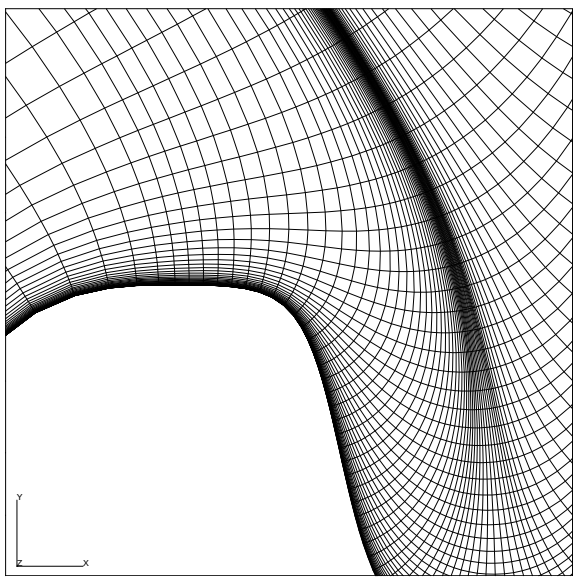

Figure 21: Detail of elliptic grid at convex part of the boundary.

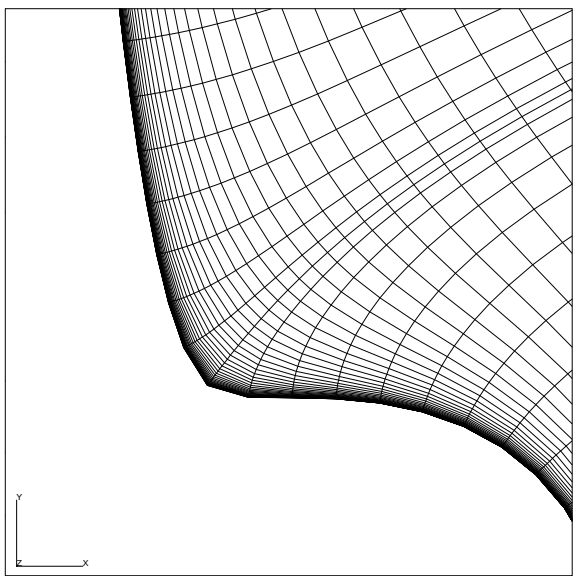

Figure 22: Detail of elliptic grid at concave part of the boundary.

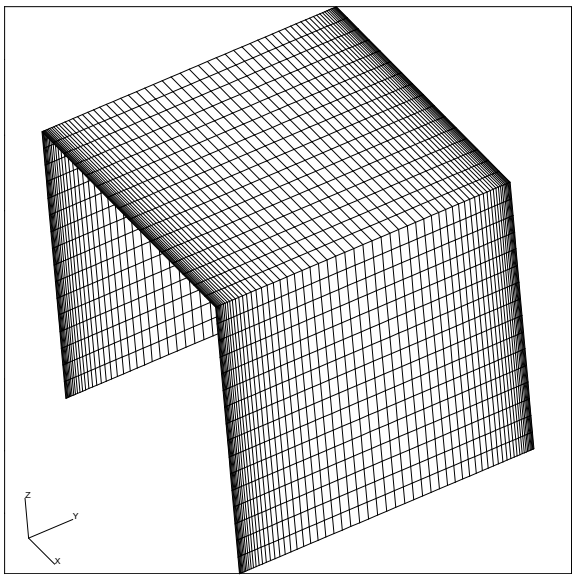

Figure 23: Initial grid used for minimal surface grid generation.

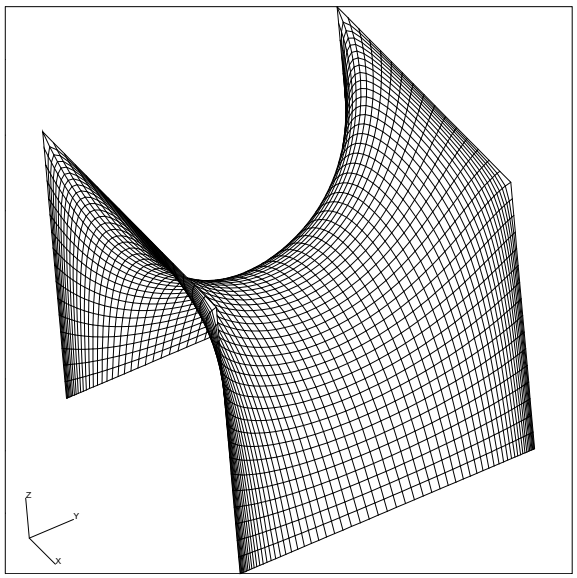

Figure 24: Minimal surface grid. Surface is a square Sherck surface.

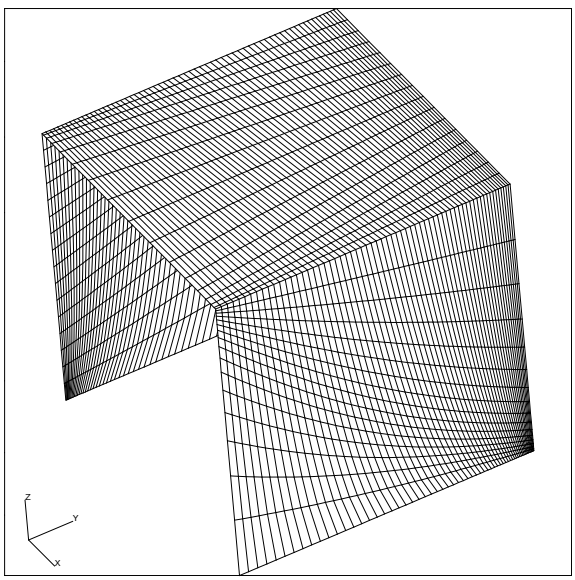

Figure 25: Initial grid used for minimal surface grid generation.

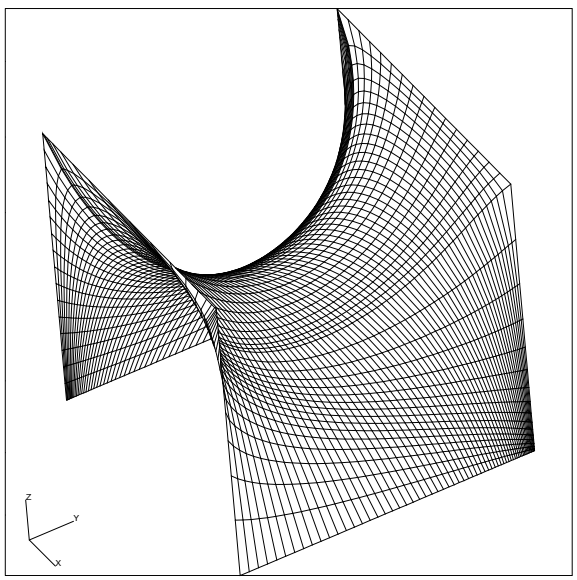

Figure 26: Minimal surface grid. Shape of surface is independent of the boundary grid point distribution.

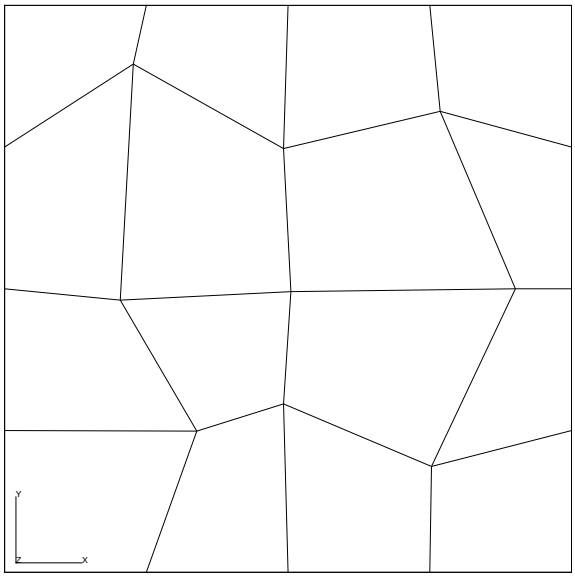

Figure 27: Surface defined by an irregular control point mesh in a unit interval.

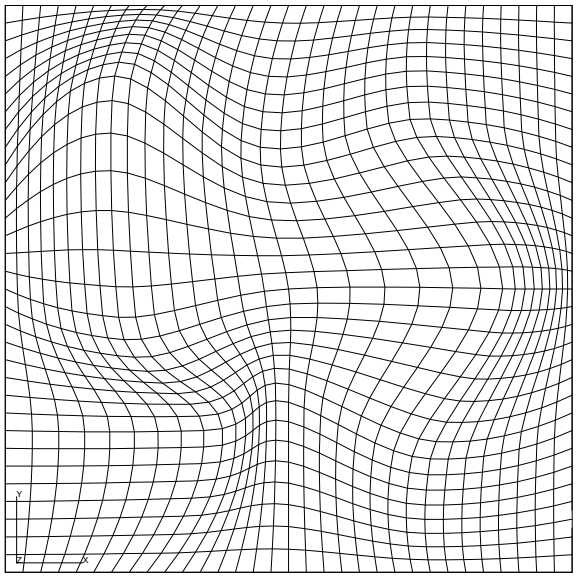

Figure 28: Initial algebraic surface grid obtained from a uniform grid in parameter space $\mathcal{P}_{uv}$.

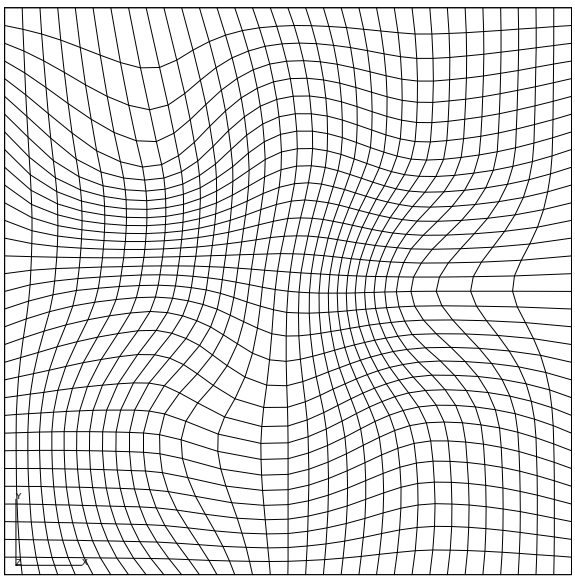

Figure 29: New grid in parameter space $\mathcal{P}_{uv}$.

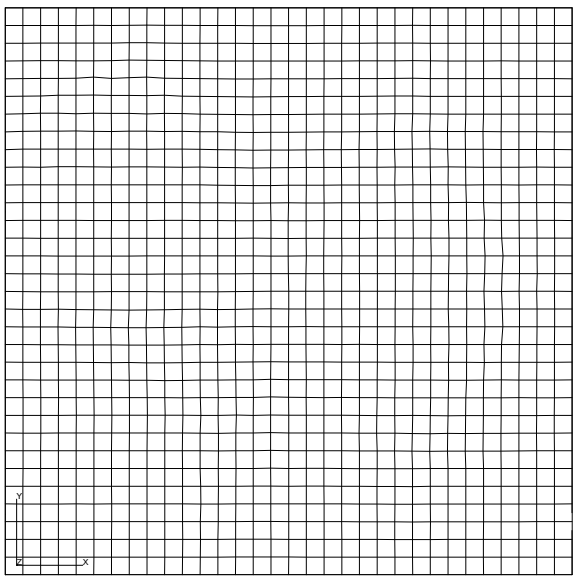

Figure 30: Corresponding elliptic surface grid. Grid is independent of parametrization.

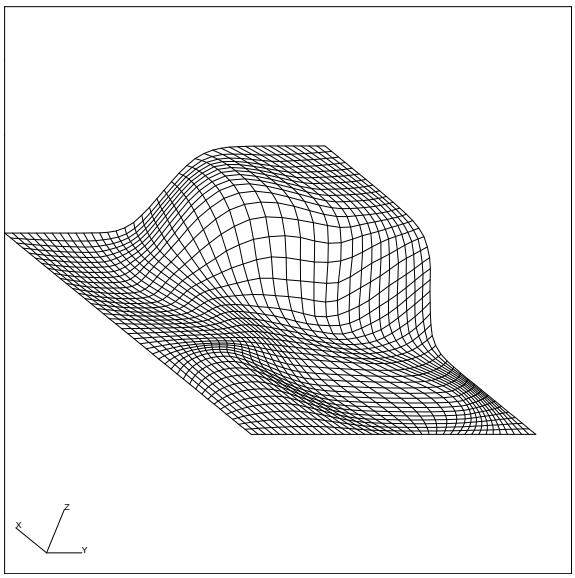

Figure 31: Irregularly distributed control point mesh on a smooth surface.

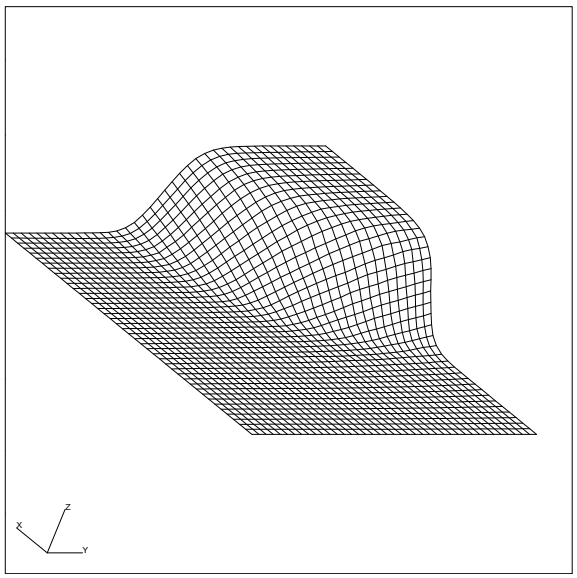

Figure 32: Elliptic grid on the surface. Grid is independent of the parametrization.

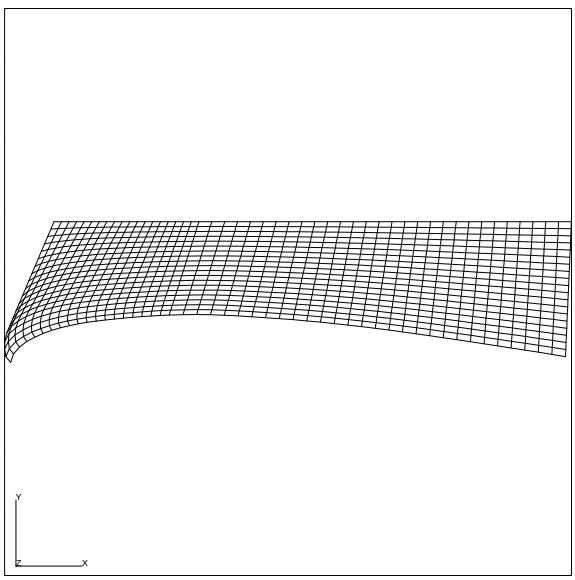

Figure 33: Control point mesh of a surface which belongs to the lower part of a wing near the intersection of a pylon.

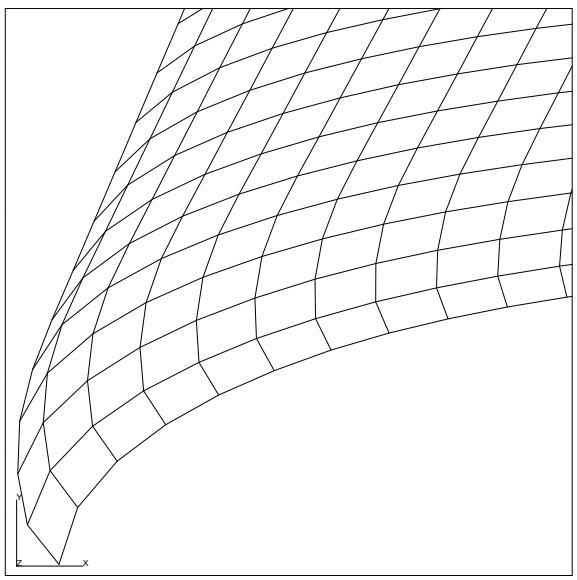

Figure 34: Detail of control point mesh.

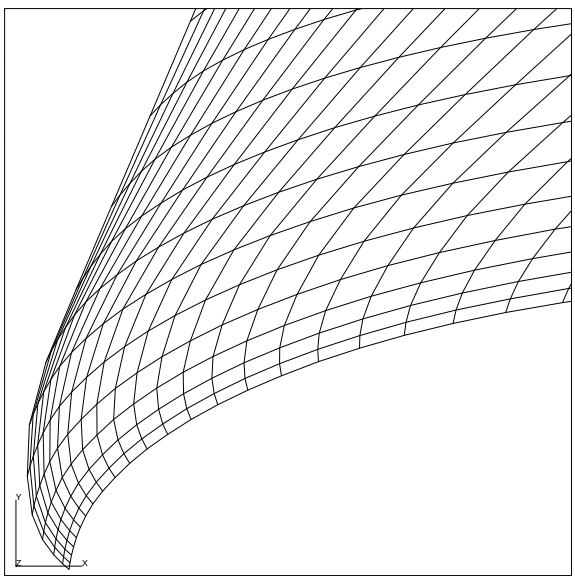

Figure 35: Detail of initial surface grid obtained by algebraic grid generation.

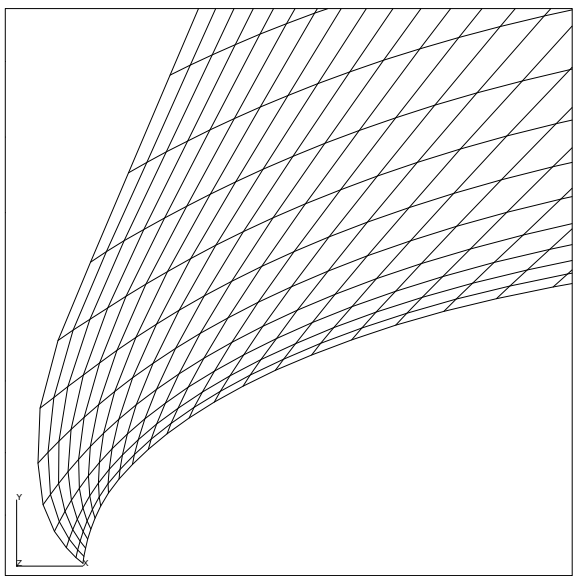

Figure 36: Detail of final surface grid obtained by elliptic grid generation.

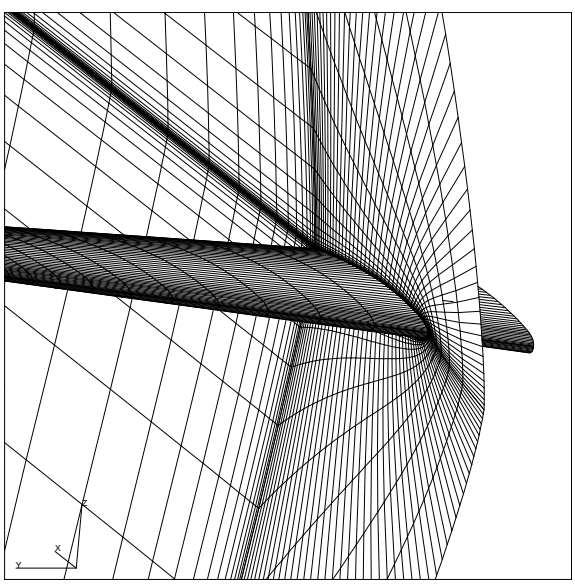

Figure 37: Vertical grid-plane intersecting the Onera-M6 wing.

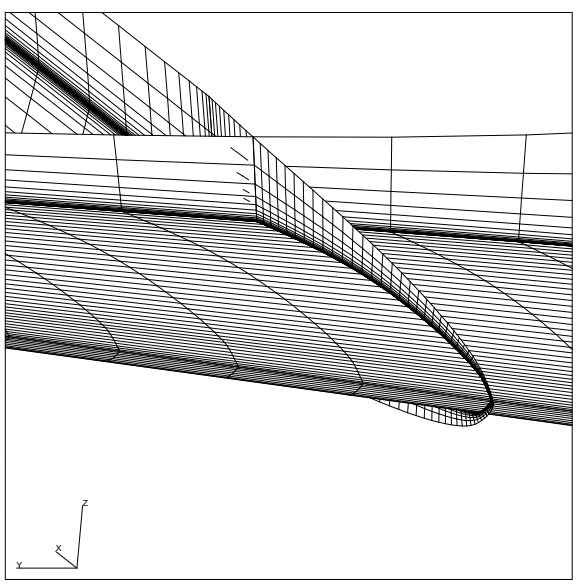

Figure 38: Onera-M6 wing with parts of vertical grid-planes.

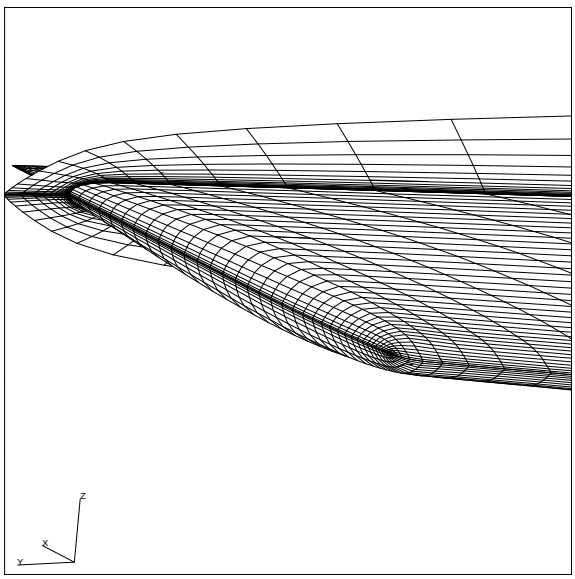

Figure 39: Wing-tip with parts of vertical grid-planes.

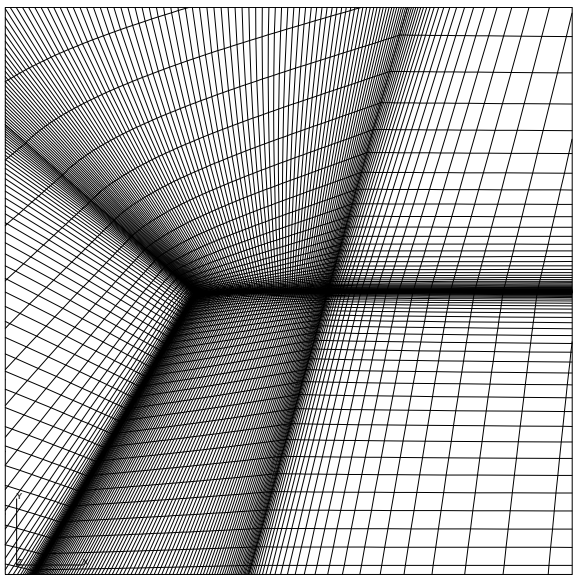

Figure 40: Horizontal grid-planes intersecting the wing.

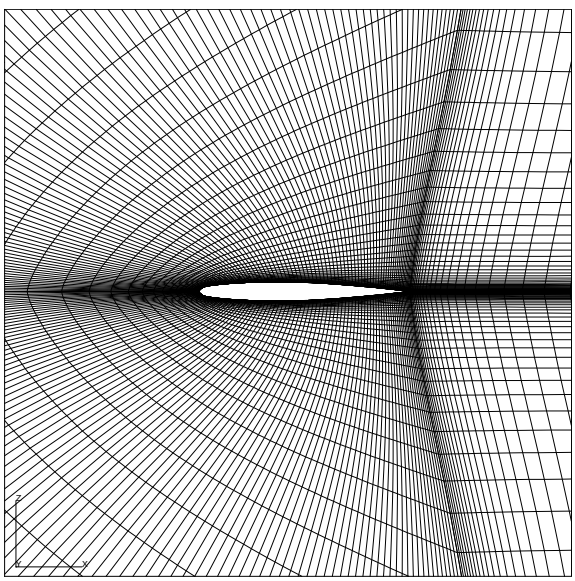

Figure 41: Grid halfway in spanwise direction.

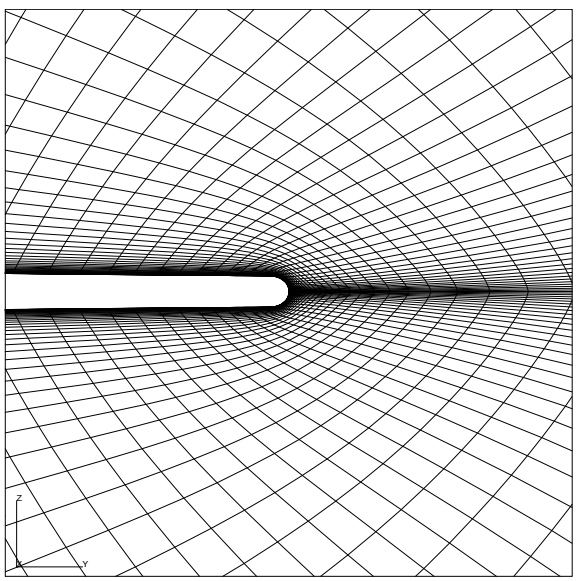

Figure 42: Grid halfway in chordwise direction.